NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

UNSTEADY PRESSURE MEASUREMENTS ON THE CASE WALL OF A TRANSONIC COMPRESSOR

by

Matt W. Rodgers

June 2003

Thesis Advisor:	Raymond Shreeve
Second Reader:	Garth Hobson

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2003	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Unsteady Pressure Measurements on the Case Wall of a Transonic Compressor	**5. FUNDING NUMBERS**
6. AUTHOR(S) Matt W. Rodgers	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Air Warfare Center Aircraft Division	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(maximum 200 words)*

The method of taking unsteady pressure measurements, on a research transonic compressor rig, was lost during the transition from the traditionally designed Vavra stage to the Sanger stage. The Sanger stage was designed using computational fluid dynamics (CFD) techniques. It required a new case wall in which the unsteady pressure sensors, due to outdated software and data acquisition system, were not initially installed. In the present study, unsteady pressure measurements were reestablished, with the installation of sensors and development of a new data acquisition and data reduction system. Data were taken at 60%, 70%, and 80% design speed. Data at 60% and 70% were compared to computational predictions and reasonable agreement was obtained.

14. SUBJECT TERMS Compressor, Transonic, Unsteady Pressure, CFD, Turbomachinery	**15. NUMBER OF PAGES** 115
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

UNSTEADY PRESSURE MEASUREMENTS ON THE CASE WALL OF A TRANSONIC COMPRESSOR

Matt W. Rodgers
Ensign, United States Navy
BSAE, Georgia Institute of Technology, 2002

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2003

Author: Matt Rodgers

Approved by: Raymond Shreeve
Thesis Advisor

Garth Hobson
Second Reader

Maximilian F. Platzer
Chairman, Department of Aeronautics and Astronautics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The method of taking unsteady pressure measurements, on a research transonic compressor rig, was lost during the transition from the traditionally designed Vavra stage to the Sanger stage. The Sanger stage was designed using computational fluid dynamics (CFD) techniques. It required a new case wall in which the unsteady pressure sensors, due to outdated software and data acquisition system, were not initially installed. In the present study, unsteady pressure measurements were reestablished, with the installation of sensors and development of a new data acquisition and data reduction system. Data were taken at 60%, 70%, and 80% design speed. Data at 60% and 70% were compared to computational predictions and reasonable agreement was obtained.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author would like to thank the staff at the Naval Postgraduate School Turbopropulsion Laboratory for all their hard work, including: Professor Raymond Shreeve, Professor Garth Hobson, Dr. Anthony Gannon, John Gibson, Doug Sievwright, and Rick Still.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The overall goal in the present turbomachinery program at the Turbopropulsion Laboratory (TPL) at the Naval Postgraduate School (NPS) is the development of optimum designs. The optimized turbomachine is achieved by designing with computational fluid dynamics (CFD) and structural analysis using the finite element method (FEM). In order to design with CFD and FEM, the techniques must be shown to be valid. The CFD technique is validated by taking flow field measurements within a turbomachine, which was designed with CFD, and comparing the computational results to the experimental. It is a process that is only now truly practical due to advances in both numerical machining techniques and computing power.

The transonic compressor used in the present study was designed using CFD techniques. It is a modern compressor, conceptually not unlike the first stage of an F119 engine. Figure 1 shows the F119 engine. Success in the present design optimization program can lead to weight reduction and improved performance in this type of engine.

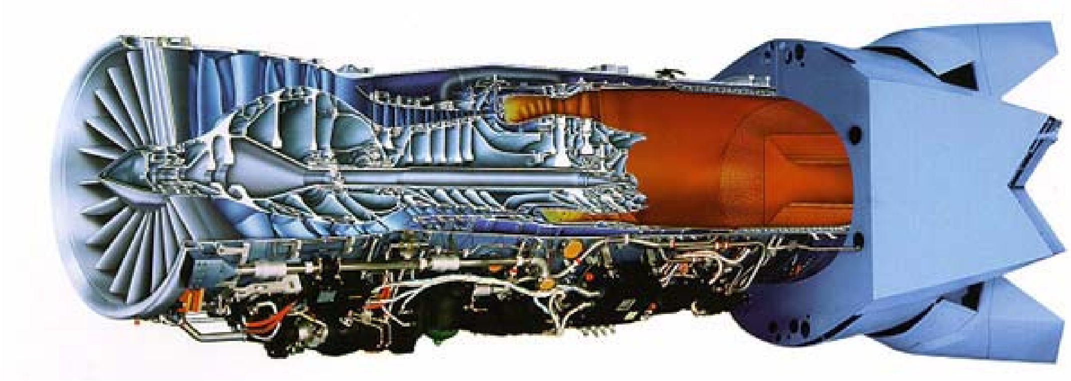


Figure 1. F119 Engine

Towards validation of design by CFD, the present study focused on unsteady pressure measurements to produce a case wall pressure contour map. Figure 2 shows the theoretical pressure map at the rotor blade tip, for the Sanger transonic compressor, at 100% design speed. The pressure map in the frame of the rotor blades is a steady pressure. From the frame of the case wall, where the pressure transducers are placed, the pressure becomes an unsteady pressure. In the present study, unsteady pressures were obtained with a single transducer and compared to the computationally predicted profile. However, hardware and software were prepared to acquire complete maps using six transducers. The present VXI-based data acquisition hardware and MATLAB analysis software completely replaced the unsteady measurement system started by Paige, [Ref. 1]. The present focus on unsteady measurements follows stage performance measurements initiated by O'Brien, [Ref. 2].

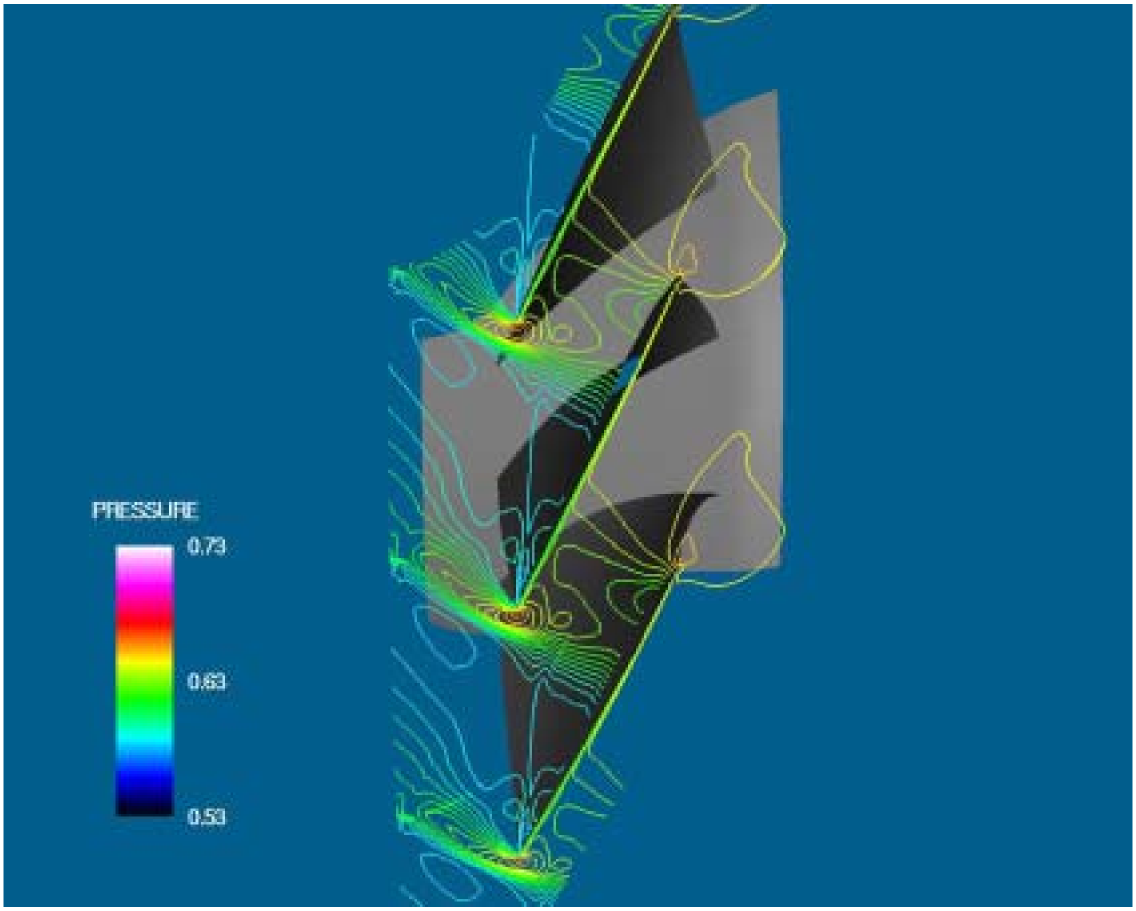


Figure 2. Theoretical Tip Pressure Map at 100% Design Speed

II. TRANSONIC COMPRESSOR

A. SANGER STAGE

The Sanger compressor stage, used in the present experiment, was designed in 1996 at the NASA Lewis Research Center, [Ref. 3]. The stage was designed using CFD techniques, while minimizing conventional empirical design methods, specifically for testing and evaluation at the Naval Postgraduate School Turbopropulsion Laboratory. Table 1 gives design specifications of the Sanger stage. The stage portrays the characteristics of the first stage of a modern fan. The design speed was set at a speed attained in previous tests of the compressor rig. The tip inlet relative Mach number is lower than most modern transonic compressors, however the blade loading is higher, which allows a pressure ratio of 1.56. Figures 3 and 4 shows the stage installed in the test rig and a sectioned drawing, respectively.

Table 1 Sanger Stage Parameters

Parameter	
Rotor Pressure Ratio	1.61
Stage Pressure Ratio	1.56
Tip Speed	1300ft/s
Design Speed	27085rpm
Design Mass Flow	17.05lb/s
Specific Mass Flow	35lbm/s-ft^2
Specific Head Rise	0.246
Tip Inlet Relative Mach Number	1.28
Aspect Ratio	1.2
Hub/Tip Radius Ratio	0.51
Rotor Inlet Ramp Angle	28.2degrees
Number of Rotor Blades	22
Number of Stator Blades	27
Tip Solidity - Rotor	1.3
Tip Solidity - Stator	1
Outside Diameter	11inches
Rotor Diffusion Factor - tip	0.4
Rotor Diffusion Factor - hub	0.47
Stator Diffusion Factor - tip	0.52
Stator Diffusion Factor - hub	0.58
Power Required	457hp

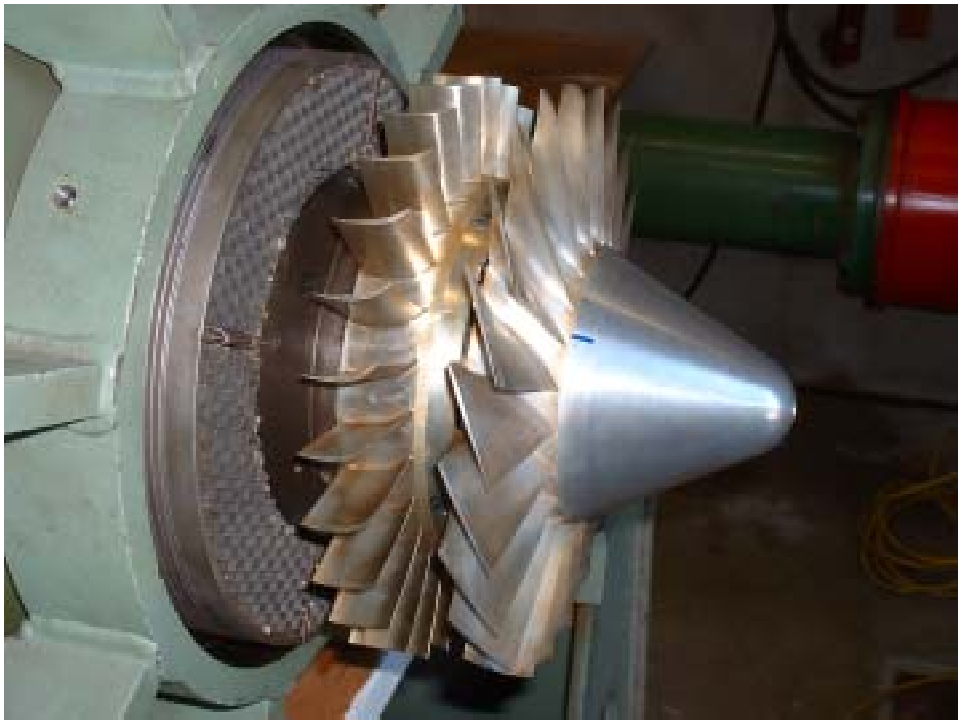


Figure 3. Transonic Compressor

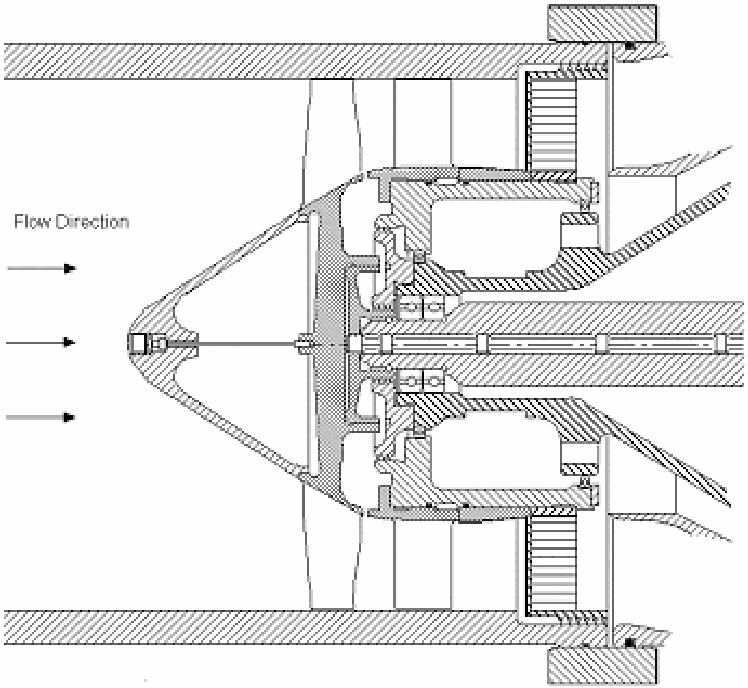


Figure 4. Transonic Compressor Sectioned View

Figure 5 shows previous data taken on the rotor and results of performance calculations, [from Ref. 4]. The data were taken with pneumatic temperature and pressure probes, torque, flow, and speed instrumentation, and represents time-averaged information. The figure is the compressor map of the Sanger stage, the circles are the measured data and the triangles are the computational predictions.

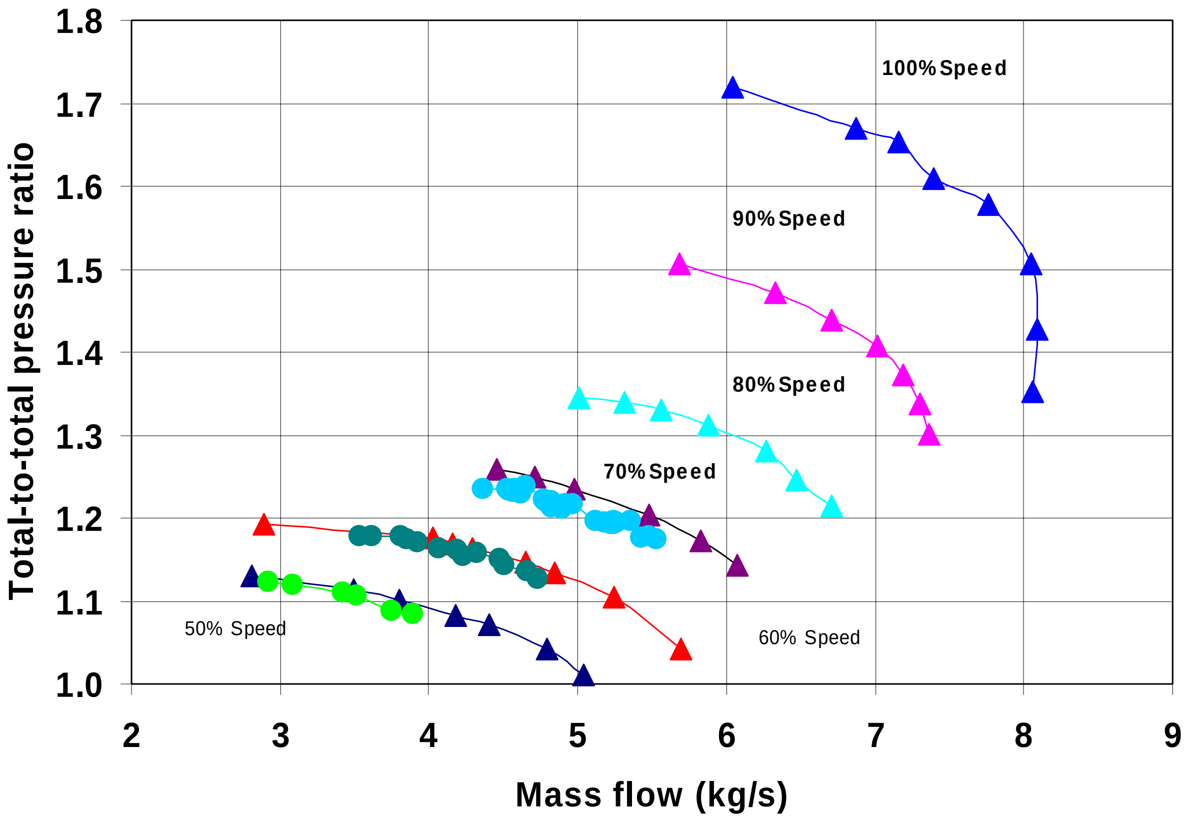


Figure 5. Compressor Map of the Sanger Stage

B. TEST RIG

The layout of the Transonic Compressor Rig and air supply system is shown in Figure 6. The rig was built in 1968 for testing a prototype transonic compressor stage. The rig and compressor were both designed by Prof. Michael Vavra, [Ref. 5]. The test compressor was driven by two opposed-rotor turbine stages, supplied by a 12-stage Allis-Chalmers axial compressor. The Allis-Chalmers supplied air up to a pressure of 30 pounds per square inch gauge, at a flow rate of 11 pounds per second. Air from the Allis-Chalmers was fed through a motor-driven valve into the turbine drive unit. The valve setting was controlled manually to control the speed of the compressor. A second high-pressure compressor, producing 150 pounds per square inch gauge, provided air to a balance piston located on the drive shaft. The balance piston controlled the axial force on the bearings in the rotor assembly. A shop compressor provided dry air for the bearing oil-mist lubrication system, and other instrument air requirements. Atmospheric air entered the transonic compressor through a throttle valve, settling chamber, inlet pipe and flow nozzle, and was exhausted back to the atmosphere.

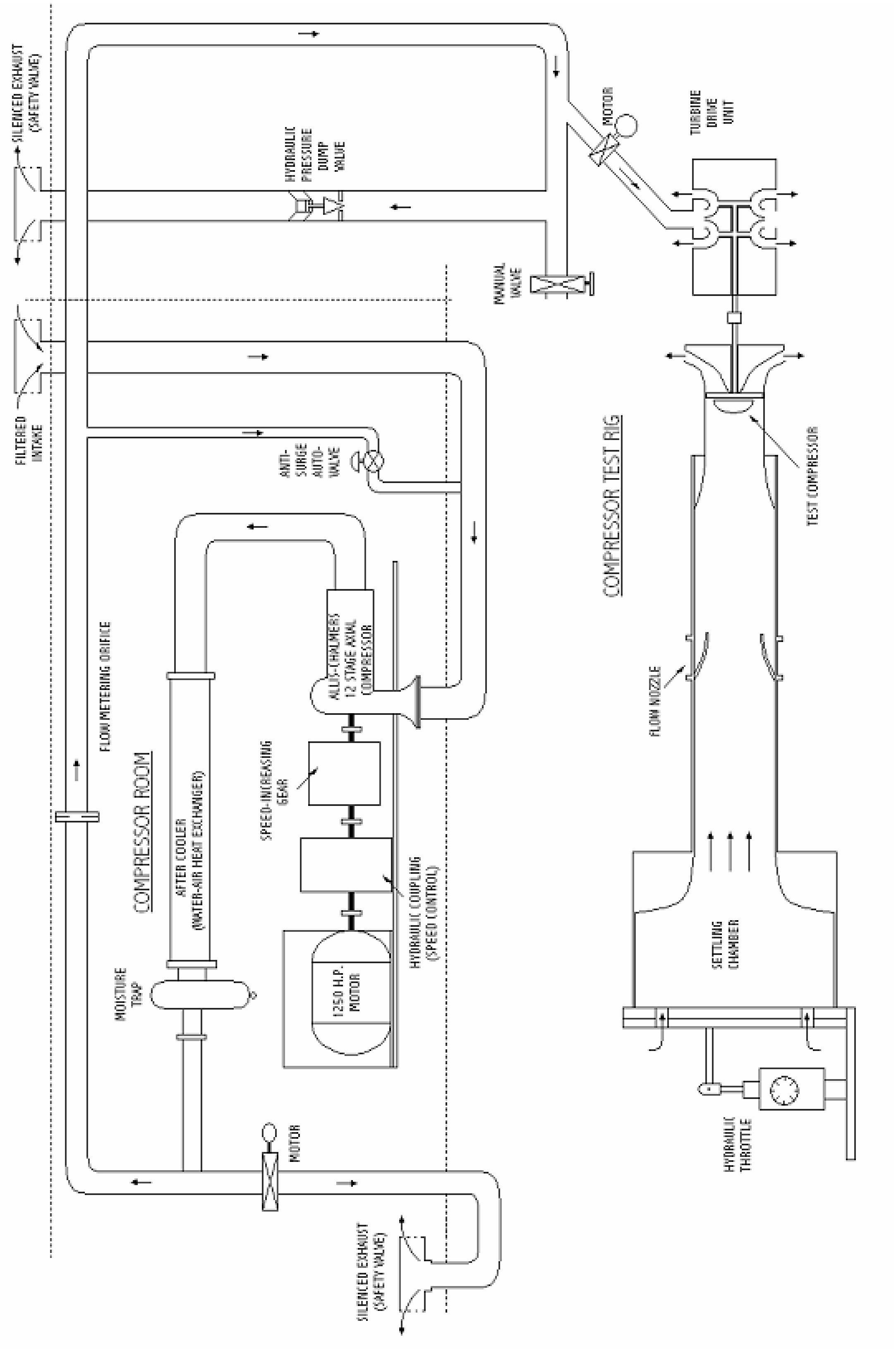


Figure 6. Rig Schematic

Figure 7 shows the Sanger stage, with the redesigned nose cone, installed in the Transonic Compressor Rig, within a Plexiglas case wall.

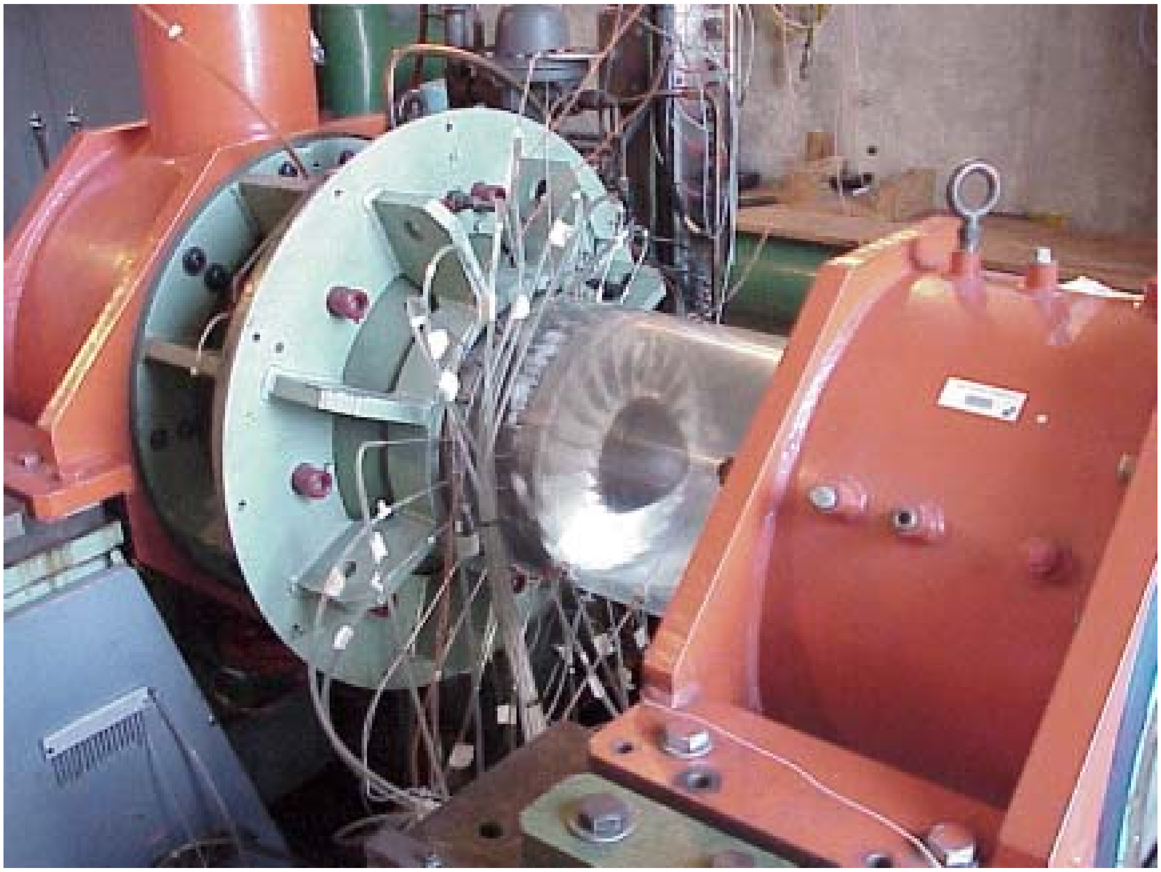


Figure 7. Test Rig

III. INSTRUMENTATION

A. PROBES

1. Kulite Pressure Transducer

A Kulite Miniature IS® Silicon Pressure Transducer was used to obtain time-resolved pressure data. The probe was a miniature, semiconductor, strain gauge transducer. The Kulite transducer incorporated a fully active four-arm Wheatstone bridge dielectrically isolated silicon-on-silicon diaphragm. A single Kulite XCQ-080-25 was chosen because of the fast response and optimal pressure range. A diagram of the Kulite is given in Figure 8. Table 2 shows a list of the factory specifications. Further information on Kulite pressure transducers is given in Ref 6.

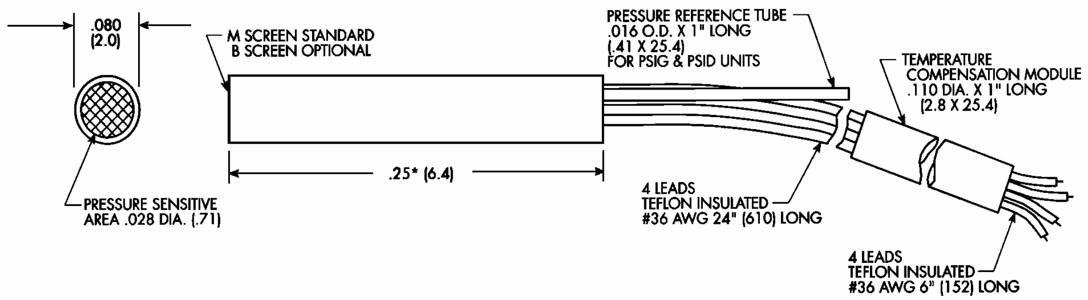


Figure 8. Kulite XCQ-080 Series Transducer

Table 2 XCQ-080-25 Factory Specifications

Input		
Pressure Range	25	psi
Over Pressure	50	psi
Burst	75	psi
Rated Electrical Excitation	10	VDC/AC
Maximum Electrical Excitation	15	VDC/AC
Input Impedance	800	Ohms
Output		
Output Impedance	1000	Ohms
Full Scale Output	100	mV
Residual Unbalance	+-3	% FSO
Non-Linearity and Hysterisis	0.1	% FS BFSL
Hysteresis	0.1	%
Repeatability	0.1	%
Resolution	Infinite	
Natural Frequency	300	kHz
Perpendicular Accel Sensitivity	0.0003	% FS/g
Transverse Accel Sensitivity	0.00004	% FS/g
Insulation Resistance	100	Megohm
Environmental		
Operating Temp Range	-65 to 250	deg F
Compensated Temp Range	80 to 180	deg F
Thermal Zero Shift	+- 1	% FS/100 F
Thermal Sensitivity Shift	+- 1	% FS/100 F

Calibration of the transducer was carried out while the compressor was running, by applying different reference pressures to the reference tube, Figure 8, and averaging the voltage recorded by the data acquisition system. Calibrating while online, alleviated the temperature dependence of the Kulite by calibrating at the running temperature of the compressor. A detailed account of the calibration procedure is given in Appendix D. MATLAB was used to average and plot the calibration curve. The code used is given in Appendix E. Figure 9 is an example of the calibration curve obtained at 60% design speed and a pressure ratio of 1.12. The Kulite shows a linear trend to reference pressure, the slope and intercept were saved for use in data reduction.

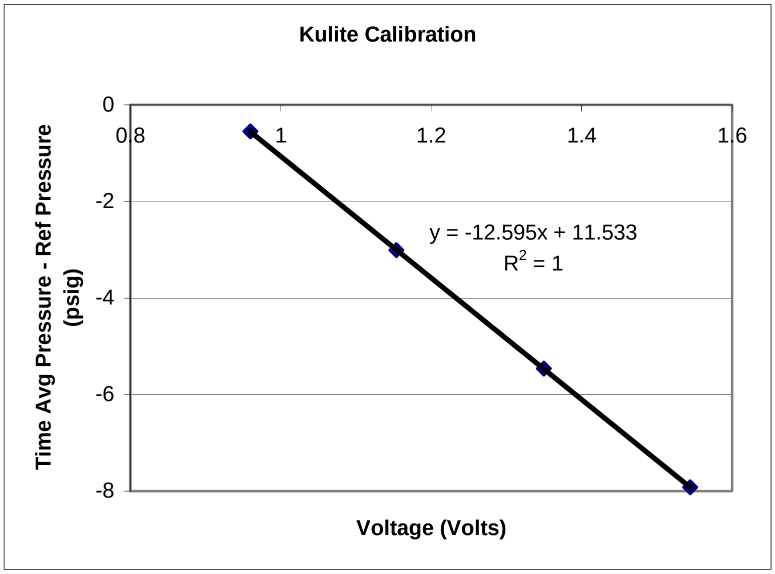


Figure 9. Kulite Calibration Curve

2. Static Pressure Measurement

A Scanivalve was used to measure the static pressure at a port located at the same axial location as the Kulite. The recorded static pressure was the time-averaged pressure used in the calibration of the Kulite pressure transducer. The pneumatically measured static pressure fixed the intercept of the calibration curve, since the time-averaged Kulite output was a measure of static-to-reference pressure difference.

B. INSTALLATION OF PROBES

The Kulite used was installed in an aluminum slug that was originally designed by Vavra for unsteady pressure measurements of the Vavra stage, [Ref. 5]. The Kulite was still in working order and permanently installed in the slug. The present installation in the case wall incorporated Vavra's installation design. Figure 10 shows the aluminum slug, designed by Vavra.

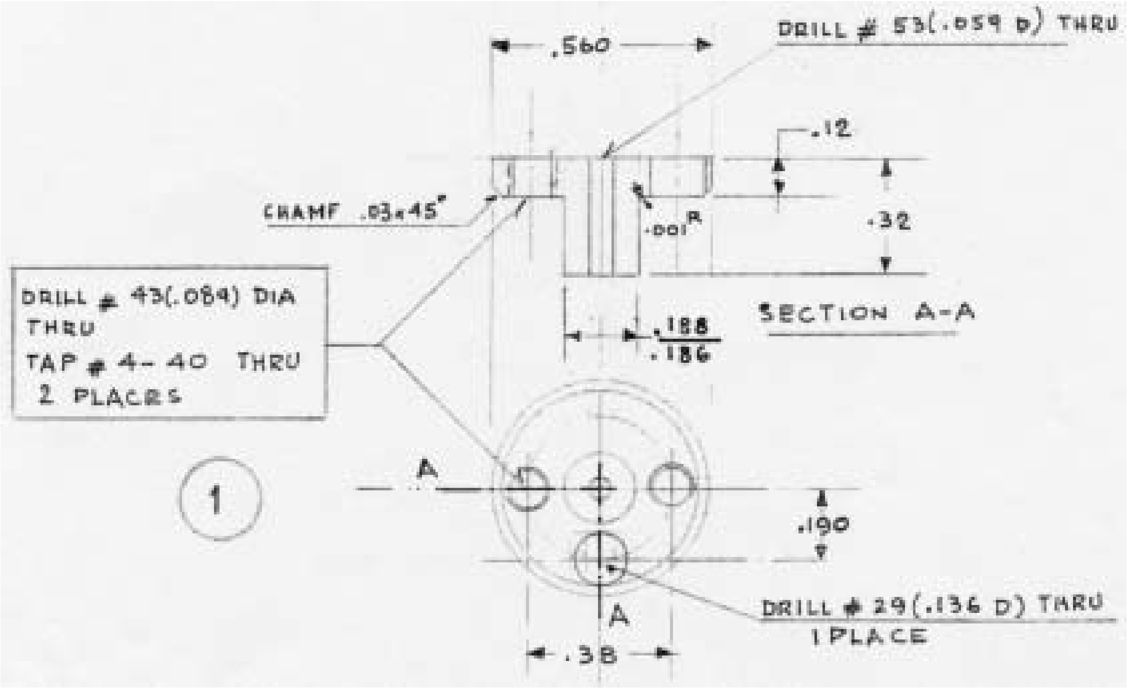
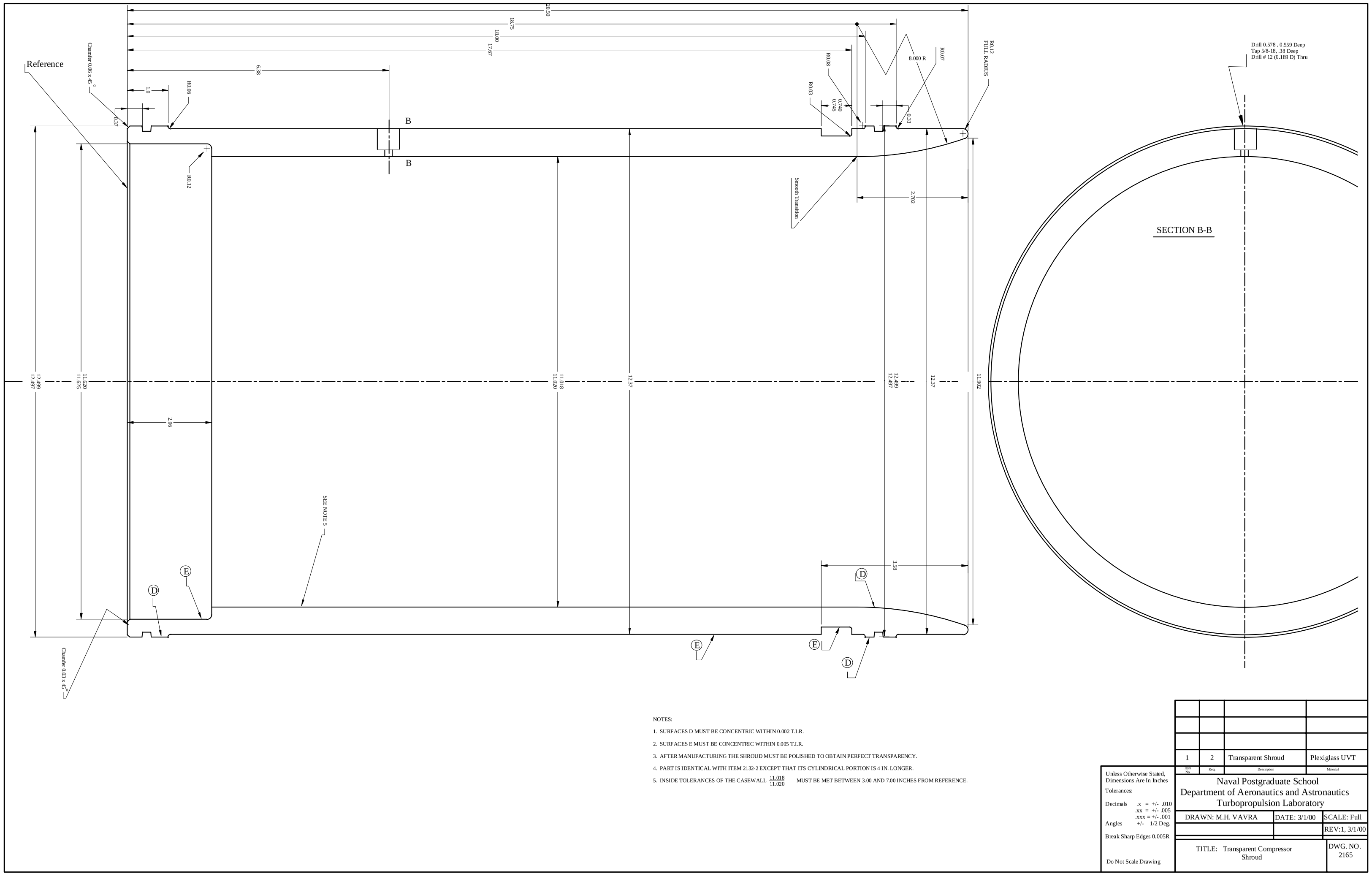


Figure 10. Kulite Mounting Design

For initial testing and evaluation, one Kulite pressure transducer was placed in the Plexiglas case wall. The Kulite probe was placed at 34% of the axial chord. The placement was chosen to be approximately where the passage shock was located at 100% design speed. Figure 11 shows the placement of the hole drilled, for the Kulite, in the case wall. The static pressure tap was also placed at an axial distance of 6.38 inches from the rear of the compressor shroud.



NOTES:

1. SURFACES D MUST BE CONCENTRIC WITHIN 0.002 T.I.R.
2. SURFACES E MUST BE CONCENTRIC WITHIN 0.005 T.I.R.
3. AFTER MANUFACTURING THE SHROUD MUST BE POLISHED TO OBTAIN PERFECT TRANSPARENCY.
4. PART IS IDENTICAL WITH ITEM 2132-2 EXCEPT THAT ITS CYLINDRICAL PORTION IS 4 IN. LONGER.
5. INSIDE TOLERANCES OF THE CASEWALL $\frac{11.018}{11.020}$ MUST BE MET BETWEEN 3.00 AND 7.00 INCHES FROM REFERENCE.

1	2	Transparent Shroud	Plexiglass UVT
Item No	Req.	Description	Material

Unless Otherwise Stated,
Dimensions Are In Inches

Tolerances:

Decimals .x = +/- .010
.xx = +/- .005
.xxx = +/- .001
Angles +/- 1/2 Deg.

Break Sharp Edges 0.005R

Do Not Scale Drawing

Naval Postgraduate School
Department of Aeronautics and Astronautics
Turbopropulsion Laboratory

DRAWN: M.H. VAVRA	DATE: 3/1/00	SCALE: Full
		REV:1, 3/1/00

TITLE: Transparent Compressor Shroud

DWG. NO. 2165

Figure 11. Kulite Placement

C. DATA ACQUISITION

The data acquisition system, used in this experiment, was installed in the NPS Turbopropulsion Laboratory for taking strain gage measurements in the spin pit. The data acquisition system is shown in Figure 12. Components installed in a ‘C’ sized VXI mainframe were interfaced to a PC, using a ‘firewire’ interface. The Kulite pressure transducer, configured as a full bridge, was connected to the Hewlett-Packard E1529A Remote Strain Conditioning Unit, [Ref. 7], via a RJ-45 cable. The RJ-45 was a typical shielded LAN cable. The HP E1529A could accomodate up to 32 strain gages, each in either a quarter, half, and full bridge configuration. An HP E1422A, [Ref. 7], controlled and set the HP E1529A bridge configuration, calibration, and self test functions. A short program written in HP Vee Pro, [Ref. 8], controlled the HP E1422A, and was used to set the HP E1529A to a full bridge configuration.

The HP E1529A provided a wideband amplified output from each strain bridge signal, via a 37-pin connector, to a HP E1433A high-speed digitizer, [Ref. 9]. The HP E1433A was an 8-channel digitizer capable of taking samples up to 196 kSa/sec. Once digitized, the data were stored on an Agilent N2216A VXI/SCSI Interface Module, [Ref. 10], containing two internal 50 Gbyte drives. The HP E1422A, E1433A, and Agilent N2216A were addressed through the HP E8404A VXI Mainframe, [Ref. 9]. Appendix C has detailed information on the data acquisition system.

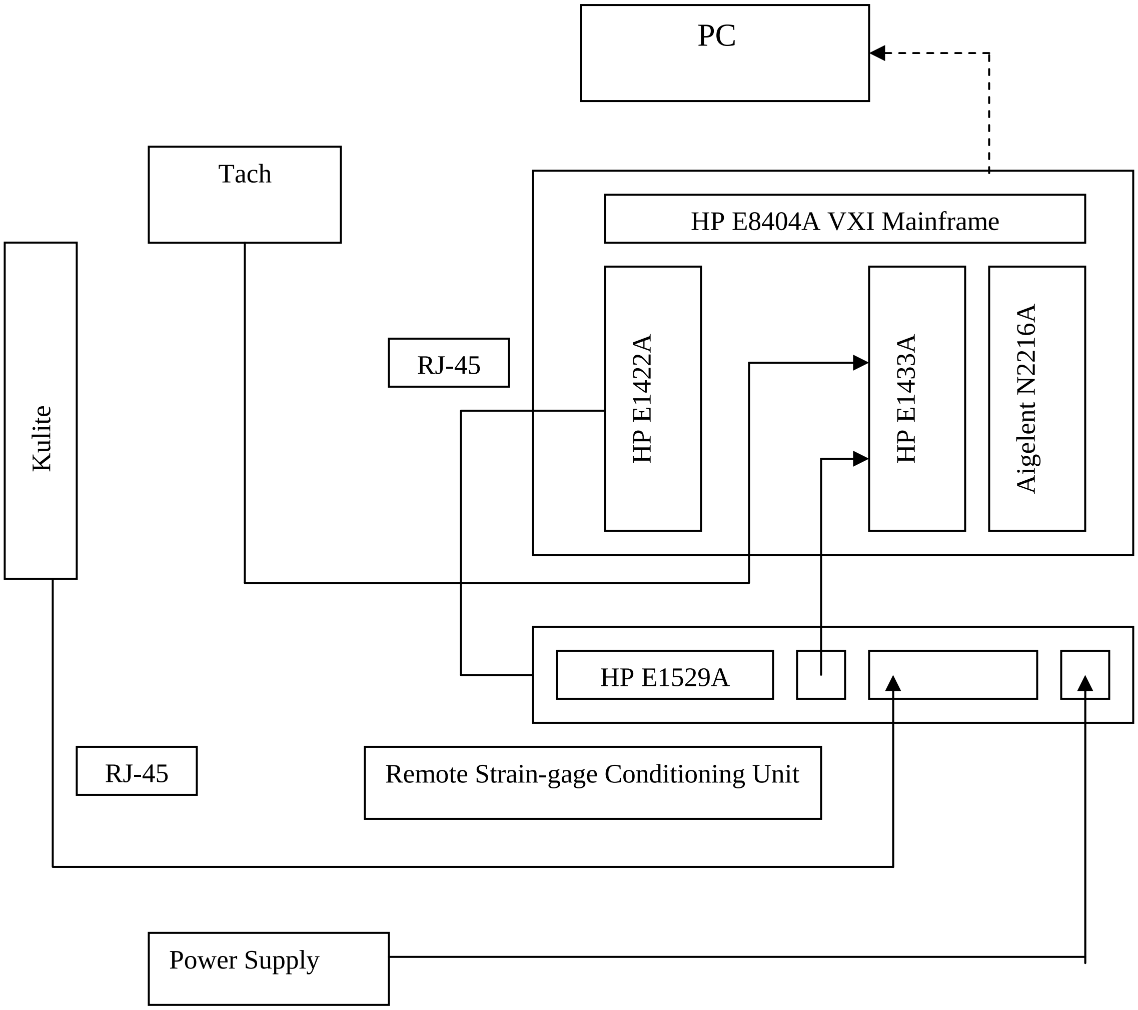


Figure 12. Data Acquisition System

1. DAC Express

Hewlett-Packard DAC Express, [Ref. 11], was used to monitor the digitized signal in real time. DAC Express set the sampling rate of the HP E1433A and recorded the digitized data to the Agilent N2216A. DAC Express can analyze up to 16 channels at a time, with the option of recording for a set amount of time. Figure 13 shows an example of the screen set-up to exhibit 11 channels in real-time, an FFT of 3 channels, tach reading, tach history, and tach signal. For dynamic measurements the sampling rate was set here to 196608 samples/sec with a sampling period of 5.09E-06 and a blocksize of 256. The length of recording was limited only by the memory size of the Agilent N2216A. A record time of .2 seconds at these parameters gave a file size of 204 kbytes.

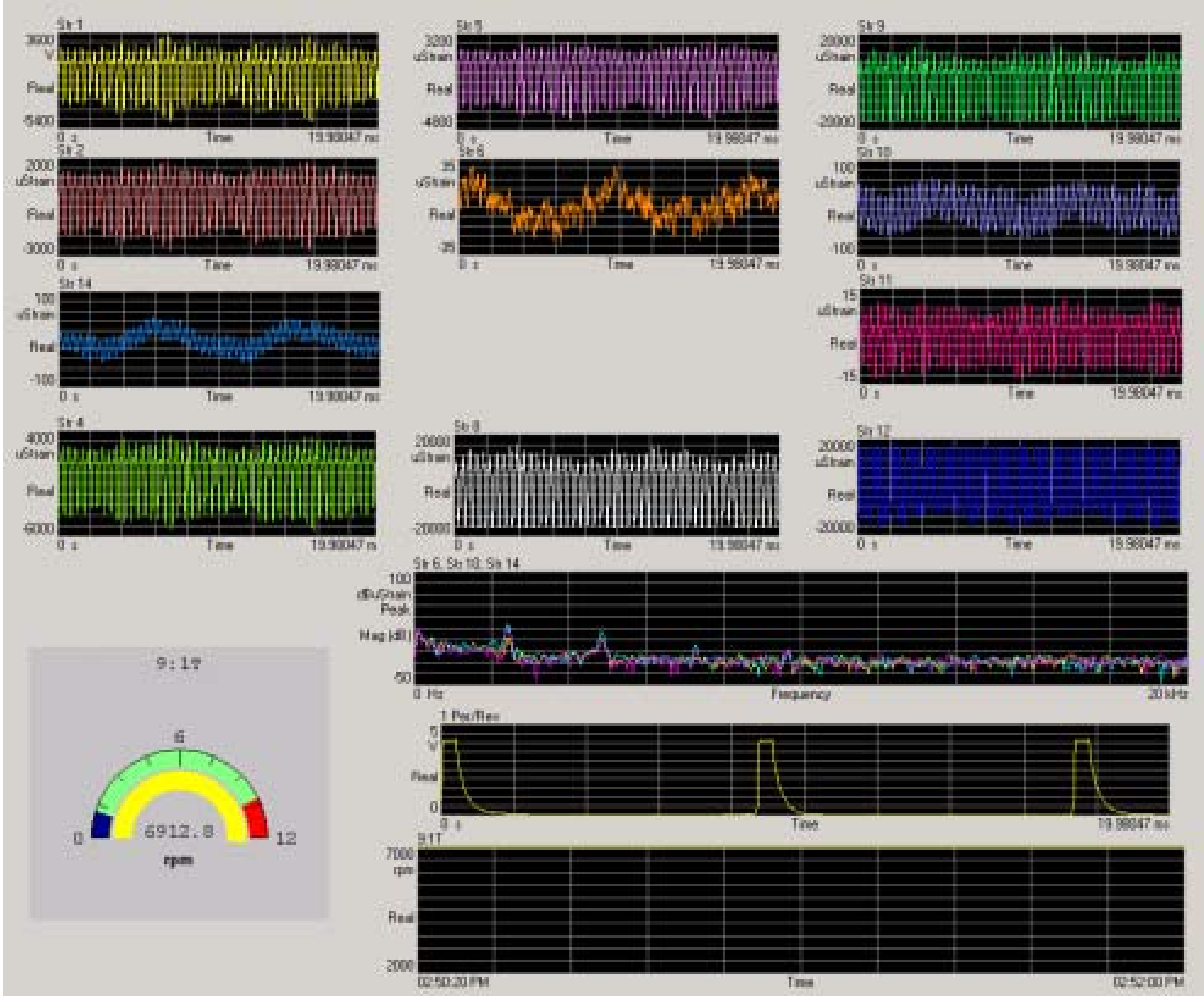


Figure 13. DAC Express Example GUI

The DAC Express GUI was configured, for the present experiment, to plot both the digitized tach voltage and the digitized unsteady voltage. A speedometer was placed on the GUI to give the speed of the compressor. Figure 14 shows the DAC Express GUI configured for this experiment. The unsteady voltage shown in the upper figure is the raw data taken from the compressor at 60% design speed, and 1.12 pressure ratio. Once the parameters were set, selecting the start button caused a set of data to be recorded. After recording, the data were exported from the N2216A to the PC as a .sdf file. A .sdf toolbox was entered into MATLAB 5.3, [Ref. 12], when DAC Express was installed. Detailed information on the DAC Express GUI is given in Appendix C.

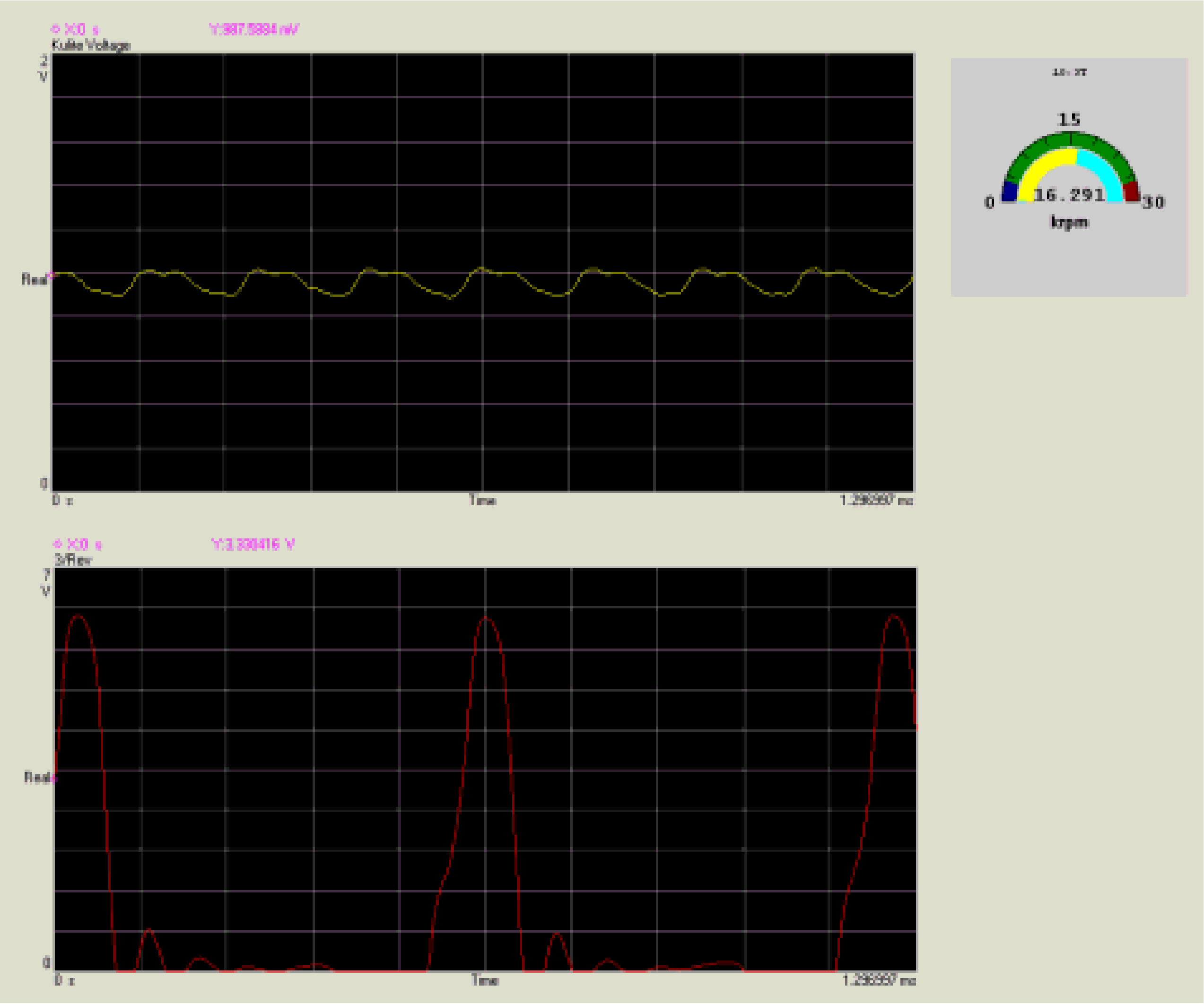


Figure 14. DAC Express GUI

THIS PAGE INTENTIONALLY LEFT BLANK

IV. TEST PROGRAM AND PROCEDURE

A. PROGRAM OF TESTS

The program of tests is summarized in Table 3. The difference in axial location of the Kulite pressure transducer was due the range of motion while removing and replacing the Plexiglas case wall. The test rig was disassembled after the first run due to a blade rub on the casing. It was concluded that faulty design of the nose cone was a factor in the rub. The rotor lip grew more than the outside of the nosecone under centrifugal loads, and separated enough to cause the nose cone to spin freely. The loose cone created an imbalance in the rotor. A redesign of the nose cone, which was shorter and locked against rotation, was built and installed in the test rig. The test rig was reassembled, with the new nose cone, and tests were resumed.

Table 3 Program of Tests

	Speed	Pressure Ratio	Axial Location
Run 1	60%	1.12	34%
	60%	1.13	34%
	60%	1.15	34%
	60%	1.16	34%
	60%	1.18	34%
Run 2	70%	1.18	38%
	70%	1.22	38%
	70%	1.24	38%
	70%	1.25	38%
Run 3	80%	1.24	38%
	80%	1.26	38%
	80%	1.27	38%
	80%	1.28	38%
	80%	1.29	38%
	80%	1.30	38%
	80%	1.31	38%
	80%	1.315	38%
	80%	1.32	38%
	80%	1.33	38%

B. PROCEDURE

1. Procedure for Taking Data

With the compressor running, four sets of data were taken for each throttle setting and corresponding pressure ratio. The four sets were at applied reference pressures of 0, 5, 10 and 15 inches of mercury, or 0, 2.456, 4.912 and 7.368 psig respectively. The applied reference pressure was manually recorded from a Wallace and Tiernan gauge with a mirrored scale graduated in .2 inches of mercury. DAC Express was set to record, at a sampling rate of ~200 kHz, the unsteady voltages for a time of .2 seconds for the 0 psig reference pressure and .02 seconds for the rest of the reference pressures. All four reference pressures were used to calibrate the Kulite but only the first reference pressure, 0 psig, was used for the unsteady pressure analysis. At 0 psig reference pressure, data were recorded for .2 seconds to record up to roughly 50 revolutions. Once recorded, the data were exported as a .sdf MATLAB format file, [Ref. 12]; this file was used for Kulite calibration and data reduction.

2. Data Reduction

A total of eight MATLAB script files were written for calibration and data analysis, including two MATLAB GUI's, [Ref. 12]. The MATLAB code is found in Appendix E.

The sequence followed in the data reduction procedure for each data set was as follows:

- Identify (encode) each data sample, with respect to position in the rotor's rotation.
- Average the revolutions to an average rotor unsteady voltage.
- Convert voltage to pressure using the slope and intercept from the calibration curve.
- Plot averaged pressure versus location around the rotor.
- Plot the standard deviation of each pressure point around the rotor.

- Plot the average blade passage from 22 blade passages.
- Plot the standard deviation of each pressure point in the averaged blade passage.
- Plot the comparison of the measured averaged blade passage to the computed blade passage.

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESULTS AND DISCUSSION

A. EXPERIMENTAL RESULTS

In the following, one set of data at one throttle setting is shown for illustration, then comparisons are given of results at different throttle settings. Figure 15 shows the rotor-averaged unsteady pressure, versus position in the rotor at a speed of 60% design speed and a pressure ratio of 1.12. Note that, the pressure ratio (absolute static pressure over upstream total pressure multiplied by gamma) is the parameter that was required to compare with the CFD results. The data were averaged from 50 revolutions, generating 2200 points over the rotor, or 100 points per blade passage. The averaged rotor was plotted with the push button, Analyze, on the Kulite Analysis GUI. The individual blade passages are seen to be similar with the exception of the 8-9 passage. This could have indicated variance in the geometry, which might have been the last blade passage that was machined, when there was no support behind the blade. Plots for the averaged rotor unsteady pressure for all data taken, are given in Appendix A.

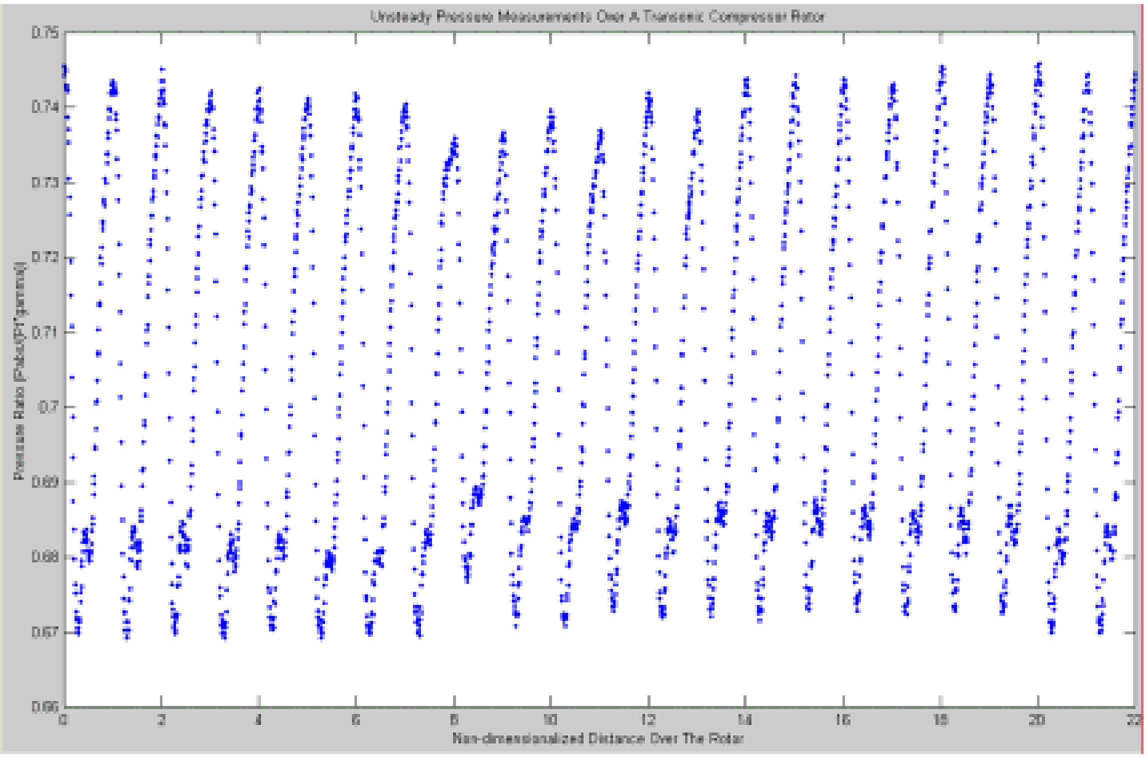


Figure 15. Unsteady Pressure over Entire Rotor

Figure 16 shows the standard deviation of the averaged rotor. The standard deviation was plotted by checking the Include Standard Deviation box, with the Analyze push button, on the Kulite Analysis GUI. The figure shows the standard deviation of all data averaged in each bin. The figure shows that the standard deviation between the revolutions was small, and that the rotor flow was relatively steady. Plots for the averaged rotor standard deviation for all data taken, are given in Appendix A.

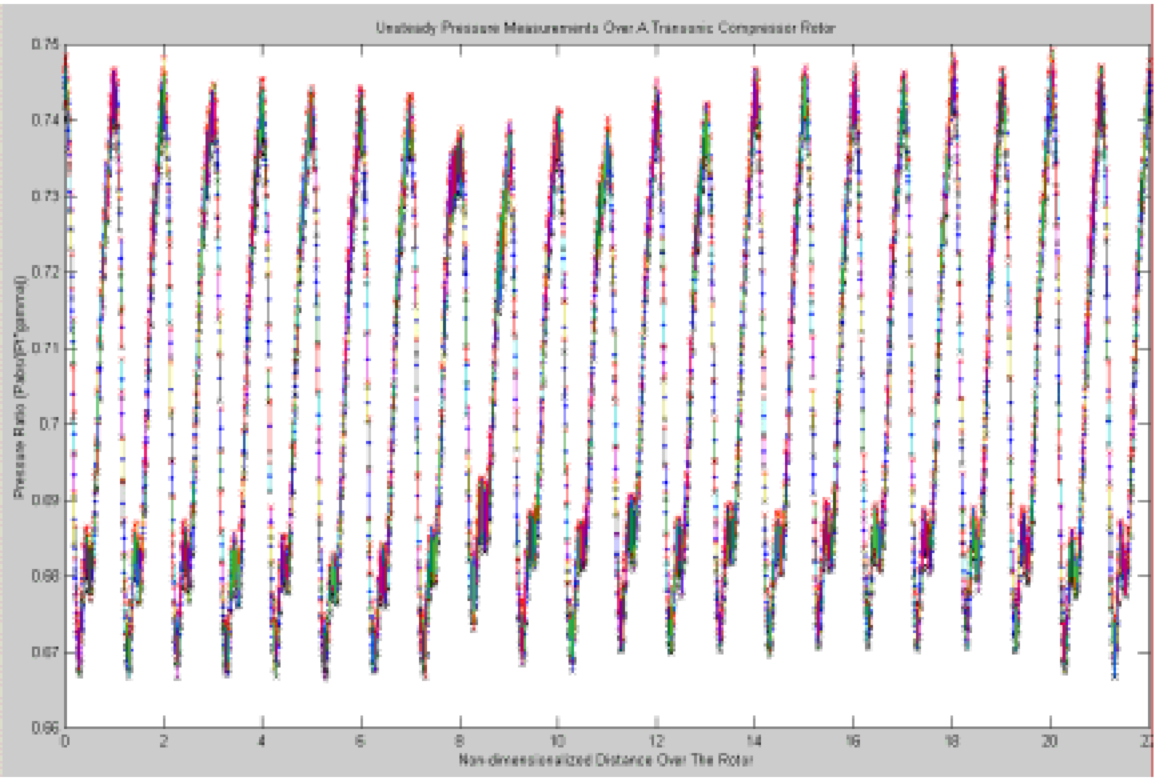


Figure 16. Standard Deviation of Entire Rotor

Figure 17 shows a plot of the unsteady pressure for the average blade passage. The average blade passage unsteady pressure was plotted with the push button, Average Blade, on the Kulite Analysis GUI. The average blade passage was plotted after the averaged rotor was analyzed. The average blade passage unsteady pressure was used to compare experimental and computational results.

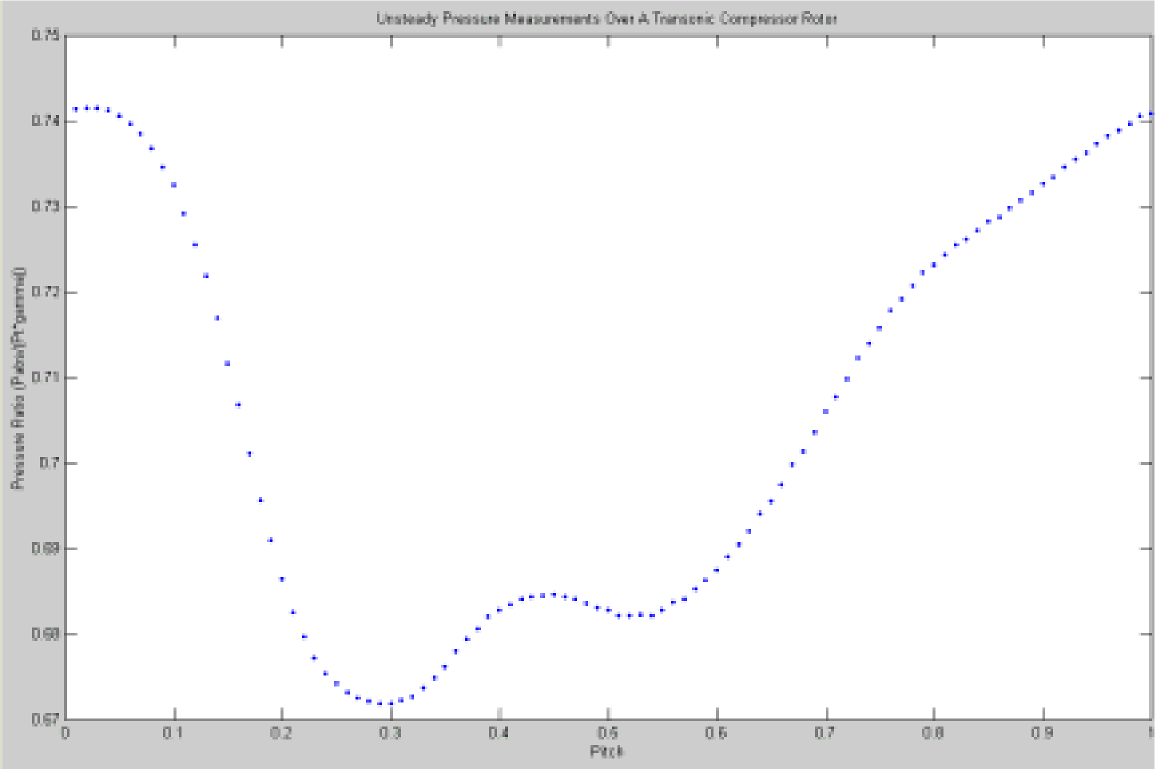


Figure 17. Unsteady Pressure of Averaged Blade Passage

Figure 18 shows the standard deviation of the averaged blade passage. The standard deviation was plotted by checking the standard deviation box, with the Average Blade push button, on the Kulite Analysis GUI. The standard deviation was a product of two averages, the averaged revolution and the averaged blade passage. The standard deviation is shown plotted on top of each pressure point for the averaged blade passage. The deviation between the blade passages is seen to be small even with the noticeable dip in pressure on the averaged rotor.

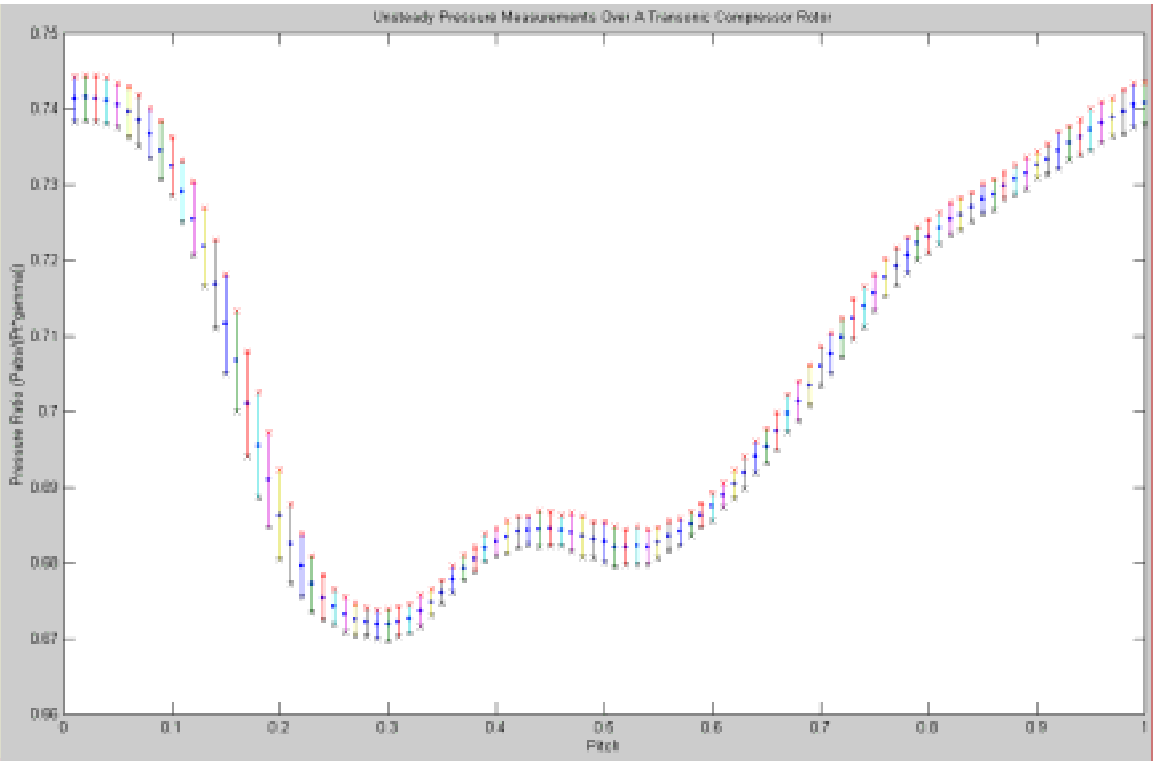


Figure 18. Standard Deviation of Averaged Blade Passage

Figure 19 shows results obtained for the measured unsteady pressure for the averaged blade passage over the range of pressure ratios. The pressure is plotted as pressure ratio versus fraction of pitch (blade spacing). The figure shows the change in the unsteady pressure as the stage pressure ratio is increased. The level of the pressure ratio increased as the compressor was throttled, which was to be expected since throttling decreased the inlet total pressure.

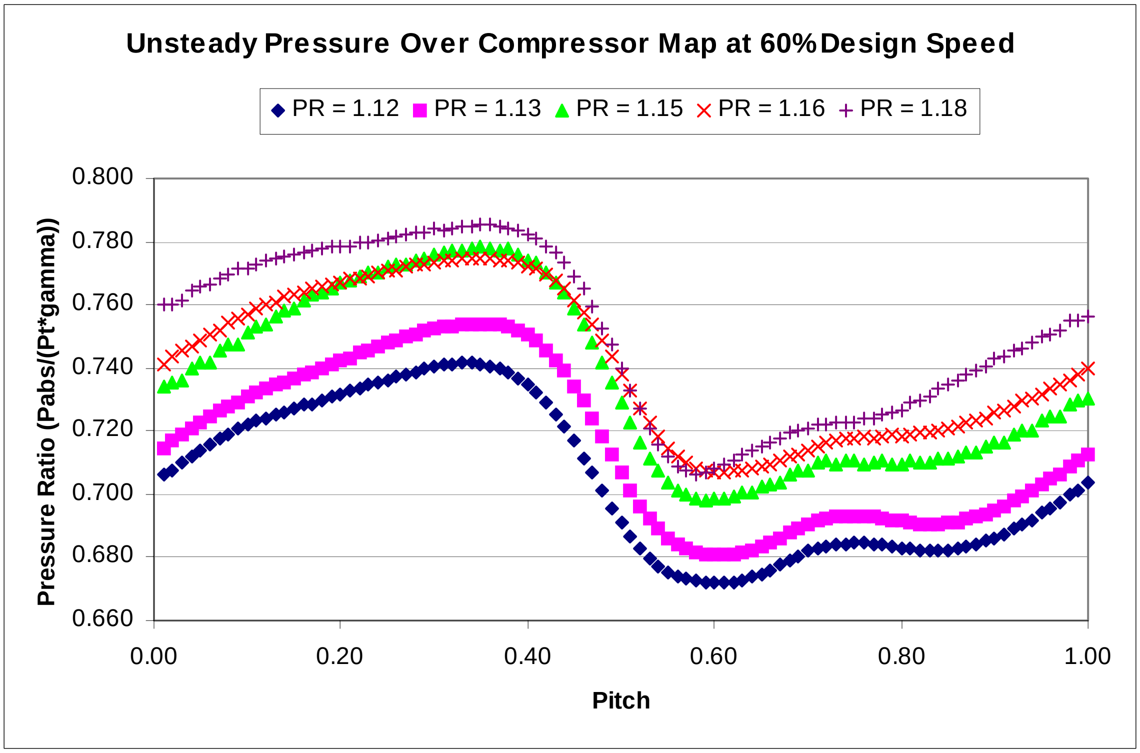


Figure 19. Unsteady Pressure Measurements over Range of Pressure Ratios at 60% Design Speed

B. COMPARISON OF EXPERIMENTAL AND COMPUTATIONAL RESULTS

Figure 20 shows a comparison of the computational results with the measured results. The figure plots the unsteady pressure at 60% design speed and 1.12 pressure ratio. The computational results were obtained using the SWIFT code developed by Chima, [Ref. 13, 14]. The results were compared using the push button, Compare, on the Kulite Analysis GUI. The pitch scale of the measured unsteady pressure was aligned with the computational scale by matching the locations of the mean of the maximum and minimum pressure values. The results taken from the compressor are qualitatively similar to the computed results. Table 4 shows the disagreement in the peak-to-peak values and in the time-averaged pressures. Both were considered to be within acceptable limits. The apparent ‘lag’ in the measured data may be a result of the gap between the Kulite pressure transducer and the blade tip. The gap was large, ~.025 inches, to spare the Kulite in case of a rub. In Figure 20, the ripple in pressure on the suction side of the blade passage is possibly the result of a tip-leakage vortex. The bars at the bottom of the figure show the width of the rotor blade and Kulite pressure transducer compared to the blade passage. Comparisons of computational and measured unsteady pressures for the 60% and 70% speed lines are shown in Appendix B.

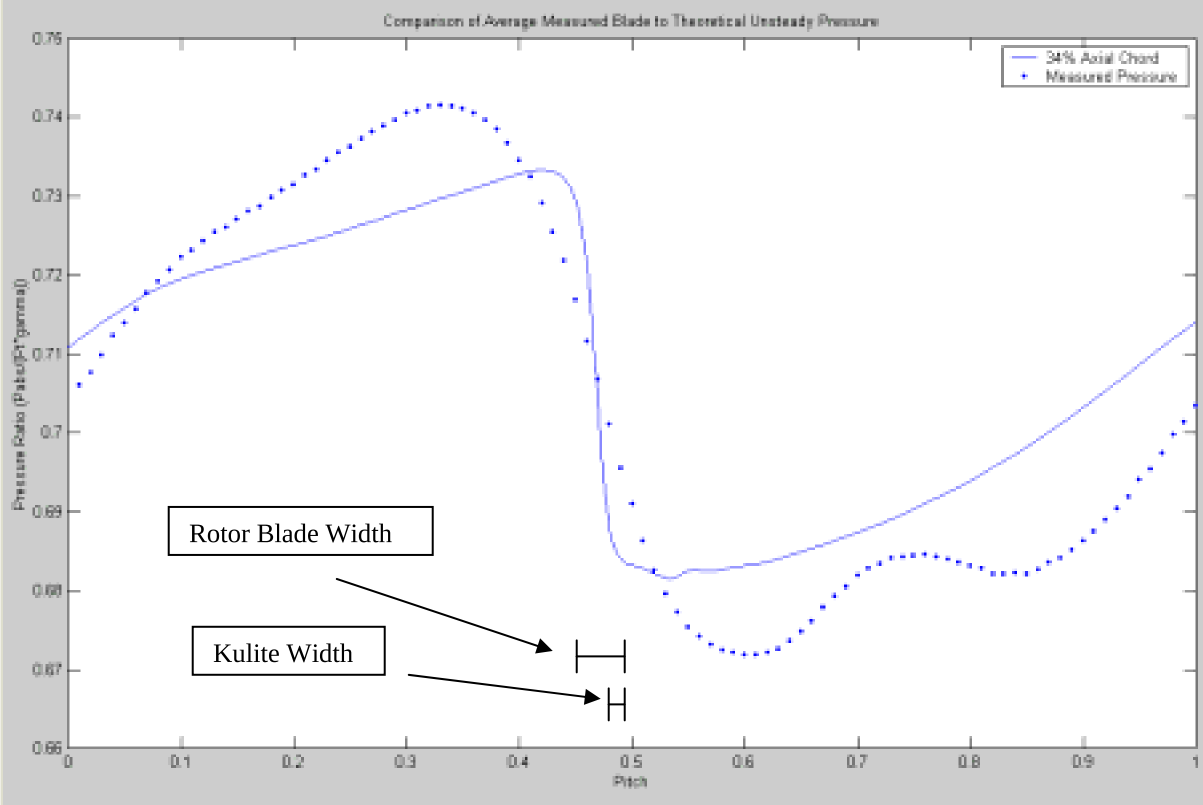


Figure 20. Comparison of Theoretical to Measured Unsteady Pressure

Table 4 Peak-to-Peak and Time Average Pressure Disagreement

	Speed	Pressure Ratio	Peak to Peak	Time-Averaged Pressure
Run 1	60%	1.12	1.2%	0.3%
	60%	1.13	1.0%	0.0%
	60%	1.15	0.8%	0.1%
	60%	1.16	0.2%	0.0%
	60%	1.18	0.6%	0.7%
Run 2	70%	1.18	2.7%	3.5%
	70%	1.22	0.7%	1.4%
	70%	1.24	0.8%	0.0%
	70%	1.25	0.8%	1.0%

VI. CONCLUSIONS

The ability to measure unsteady pressures in the transonic compressor test rig was successfully reestablished, using current VXI hardware and software generated using MATLAB. The initial data gathered from the compressor, when reduced by averaging over 50 revolutions, was smooth, with small deviation. The comparison of experimental with computational results was qualitatively favorable. The redesign and manufacture of the nose cone was required for the compressor to reach 100% design speed. Measurements with the single Kulite pressure transducer need to be taken at 90% and 100% design speed; and a full pressure contour map is needed for further assessment.

For the initial study, to generate and demonstrate the required procedures, just one Kulite was placed in the Plexiglas case wall. To derive the pressure map from unsteady pressure six Kulite pressure transducers are planned, one upstream, one downstream, and four across the blade tip. A steel case wall was designed and built for the compressor along with provision for the installation of six Kulite pressure transducers. Information on the steel case wall and Kulite installation design is given in Appendix F. Installation of the steel case wall is planned when the compressor has operated at 100% design speed in the Plexiglas case wall.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A: EXPERIMENTAL DATA

A. 60% DESIGN SPEED 34% AXIAL CHORD

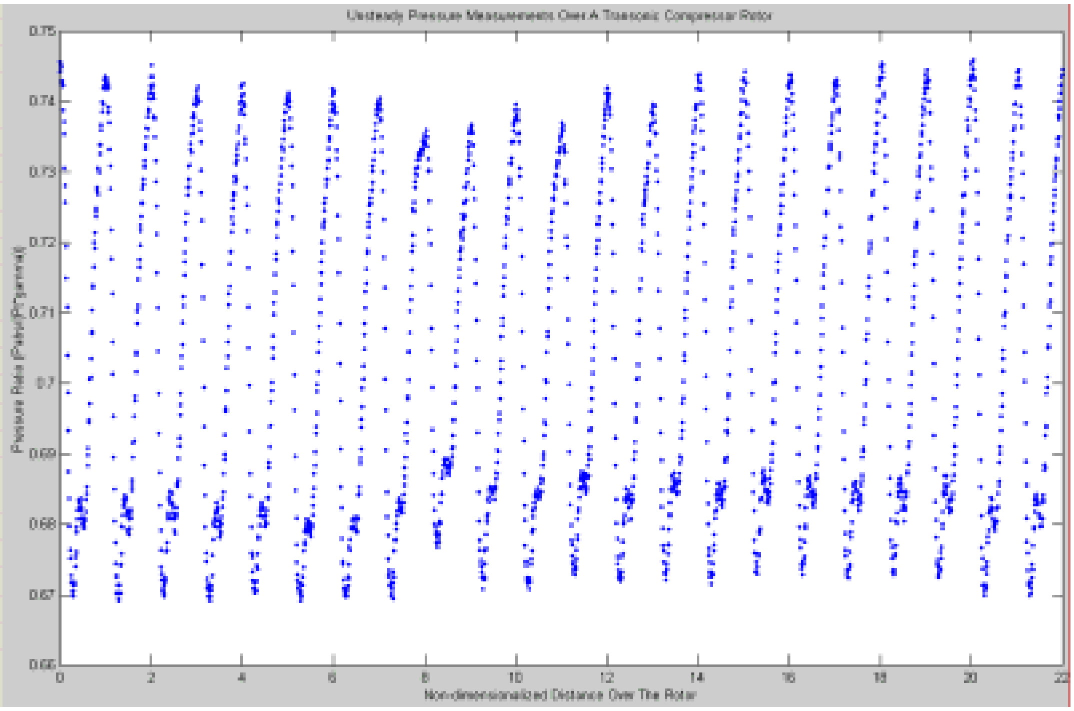


Figure 21. Unsteady Pressure over Rotor at 1.12 Pressure Ratio

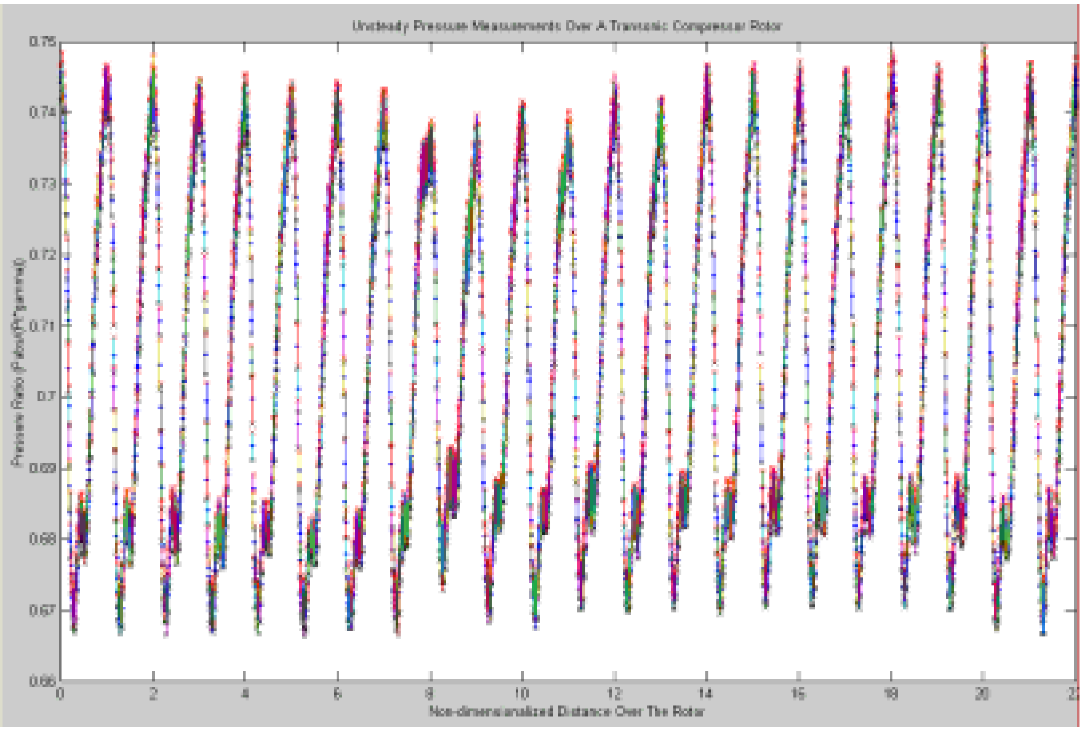


Figure 22. Unsteady Pressure over Rotor at 1.12 Pressure Ratio: Standard Deviation

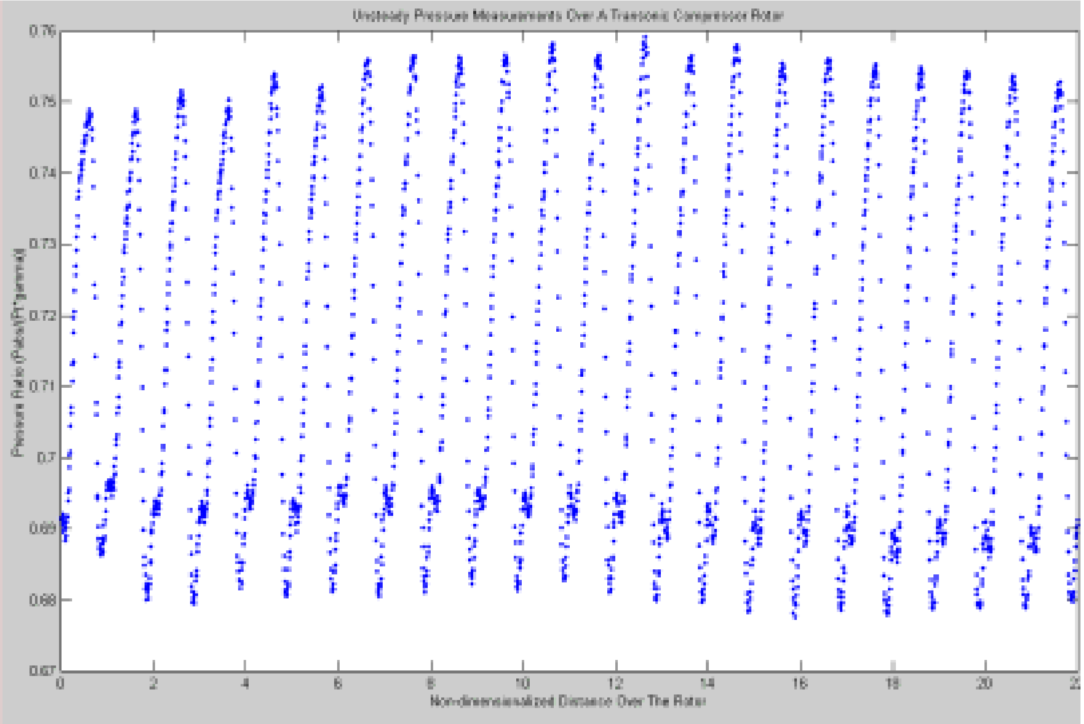


Figure 23. Unsteady Pressure over Rotor at 1.13 Pressure Ratio

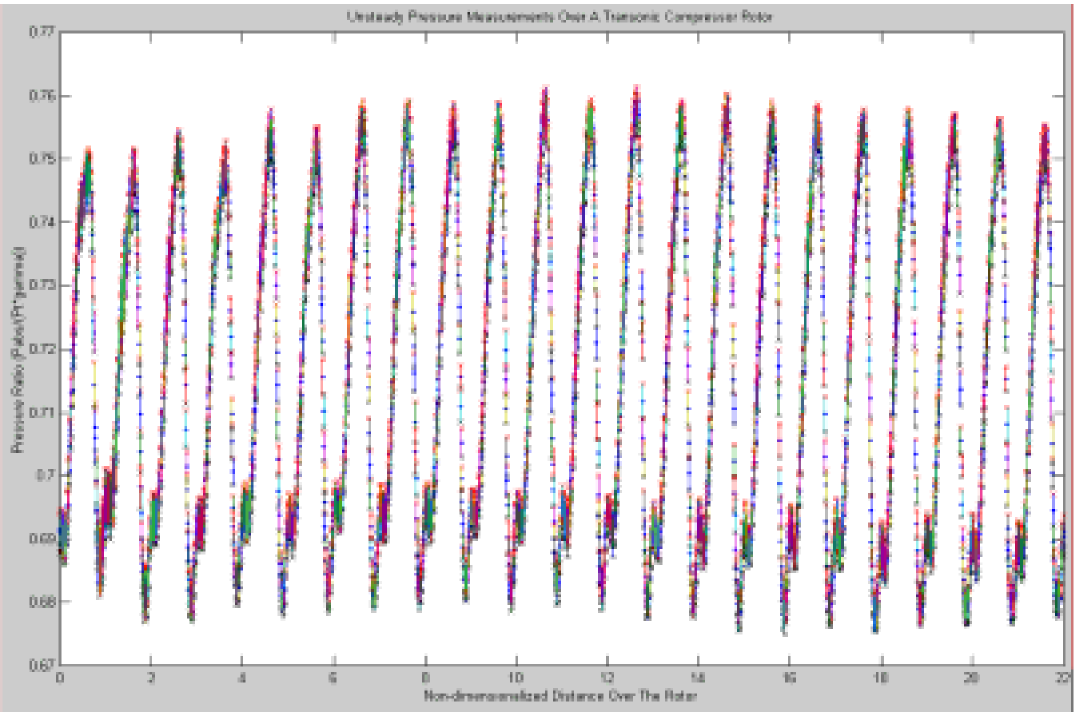


Figure 24. Unsteady Pressure over Rotor at 1.13 Pressure Ratio: Standard Deviation

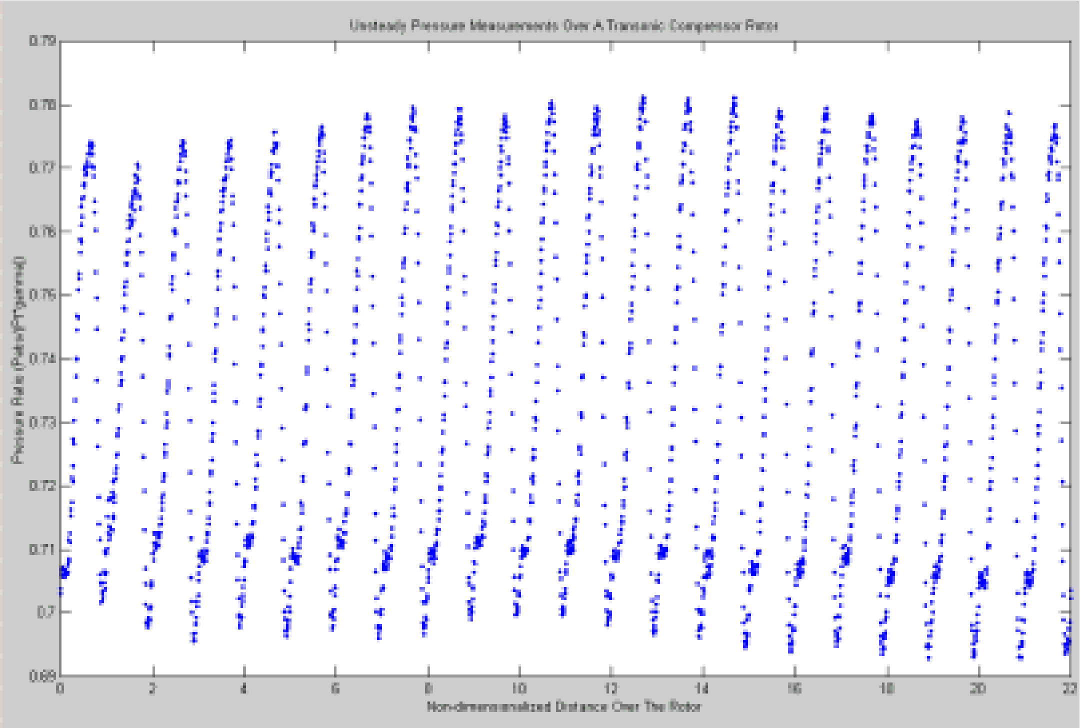


Figure 25. Unsteady Pressure over Rotor at 1.15 Pressure Ratio

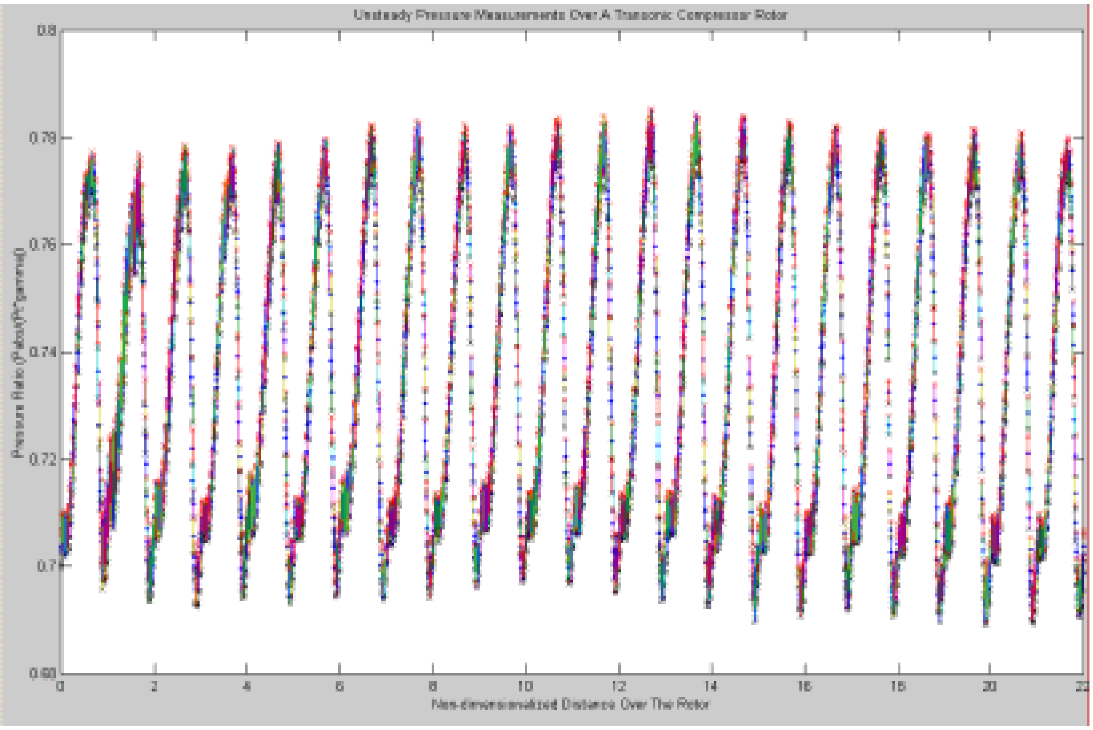


Figure 26. Unsteady Pressure over Rotor at 1.15 Pressure Ratio: Standard Deviation

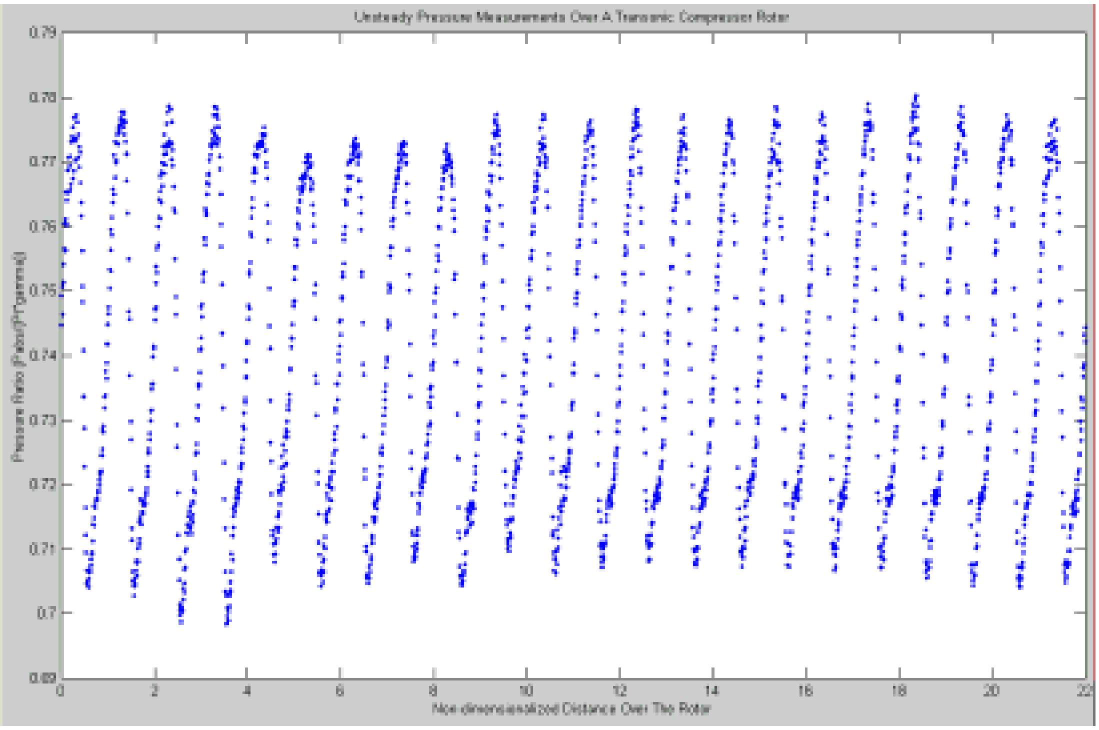


Figure 27. Unsteady Pressure over Rotor at 1.16 Pressure Ratio

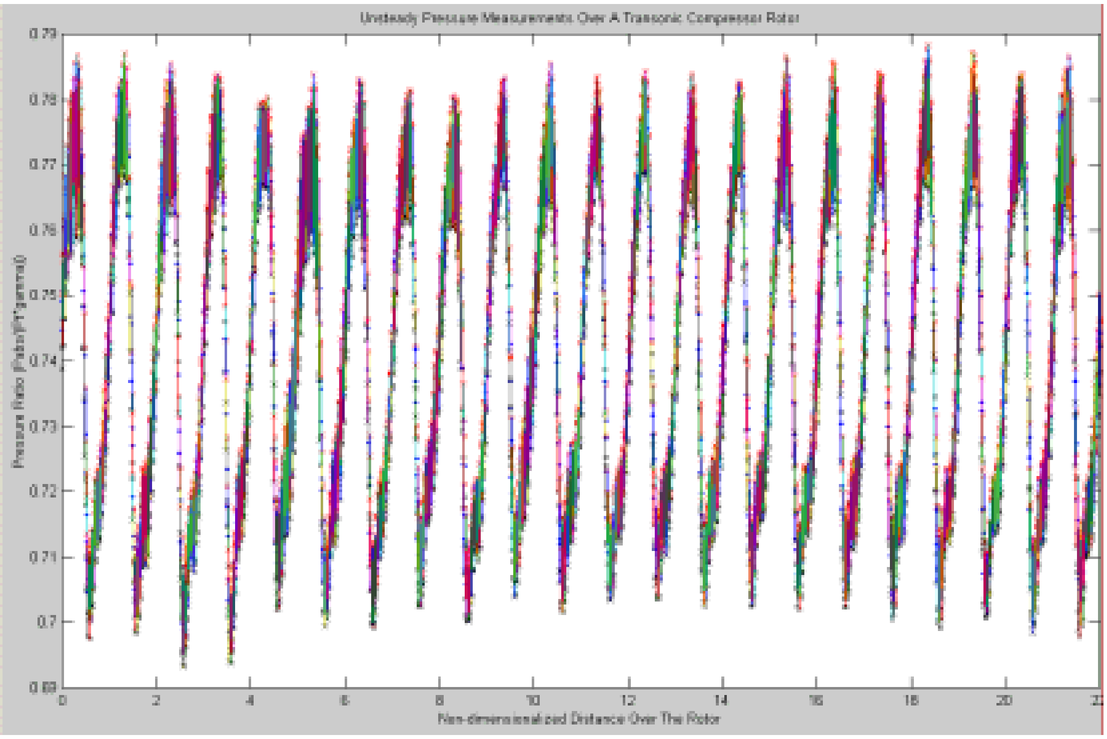


Figure 28. Unsteady Pressure over Rotor at 1.16 Pressure Ratio: Standard Deviation

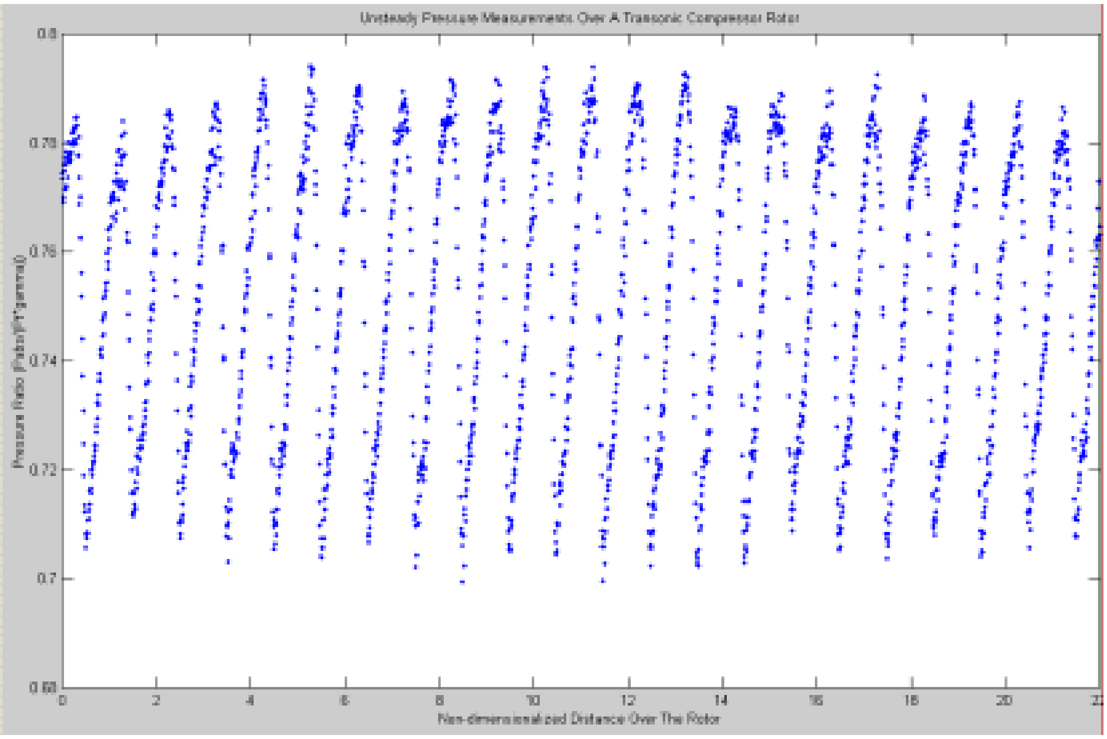


Figure 29. Unsteady Pressure over Rotor at 1.18 Pressure Ratio

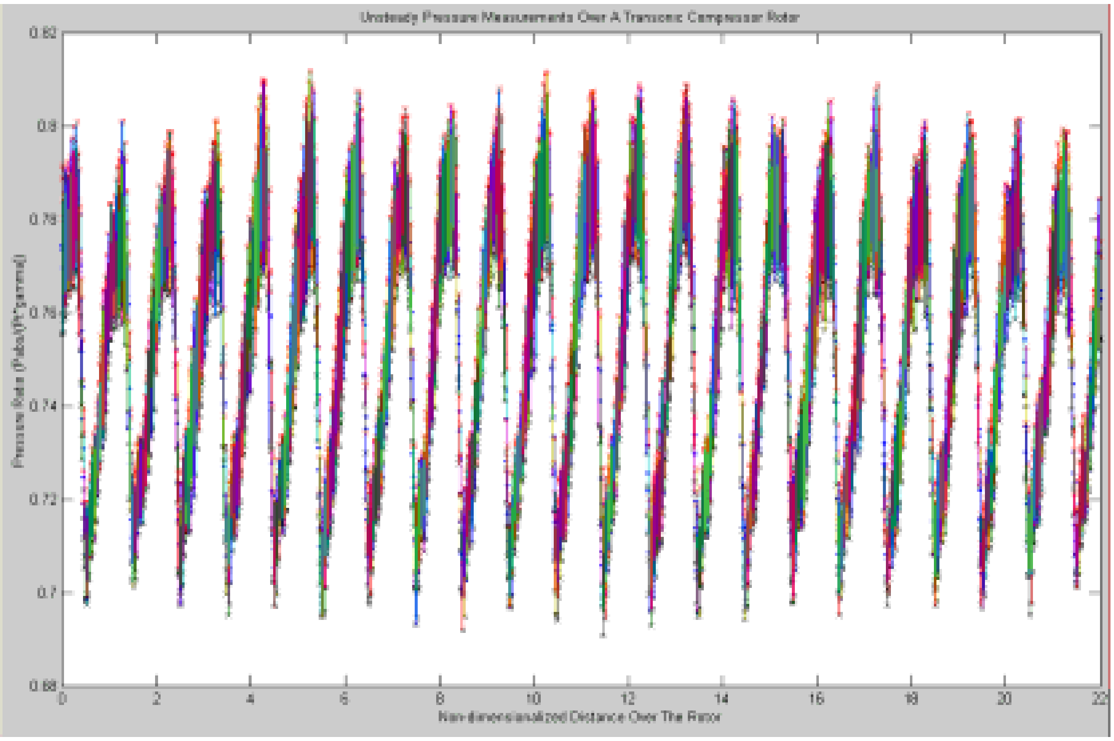


Figure 30. Unsteady Pressure over Rotor at 1.18 Pressure Ratio: Standard Deviation

B. 70% DESIGN SPEED 38% AXIAL CHORD

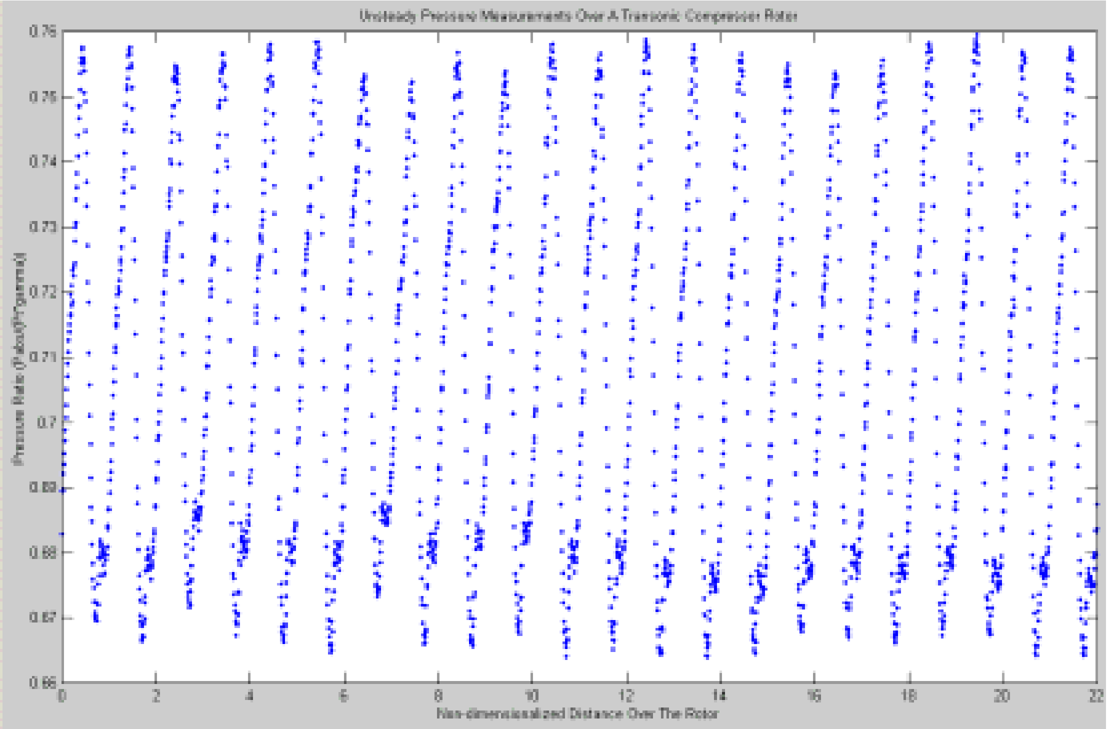


Figure 31. Unsteady Pressure over Rotor at 1.18 Pressure Ratio

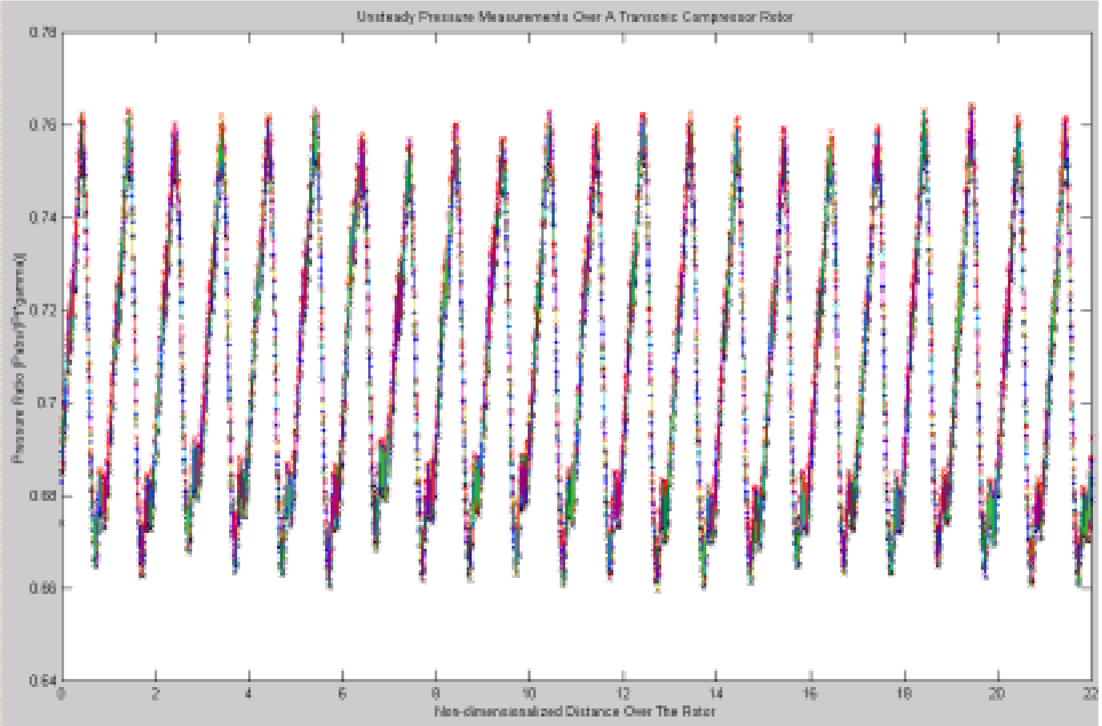


Figure 32. Unsteady Pressure over Rotor at 1.18 Pressure Ratio: Standard Deviation

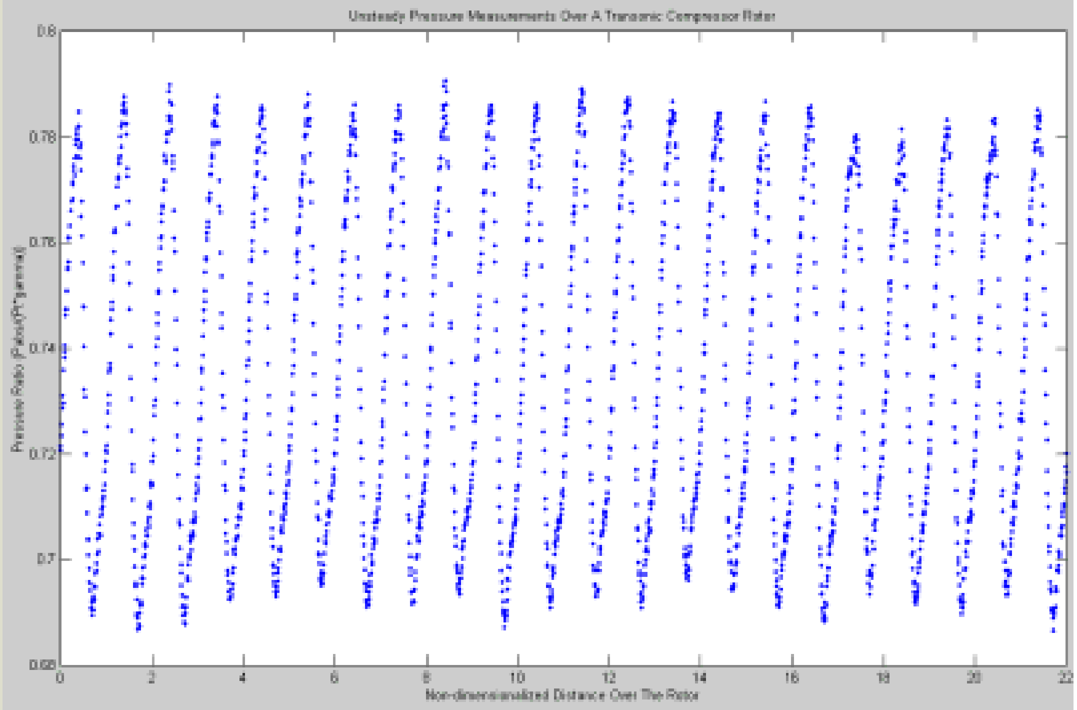


Figure 33. Unsteady Pressure over Rotor at 1.22 Pressure Ratio

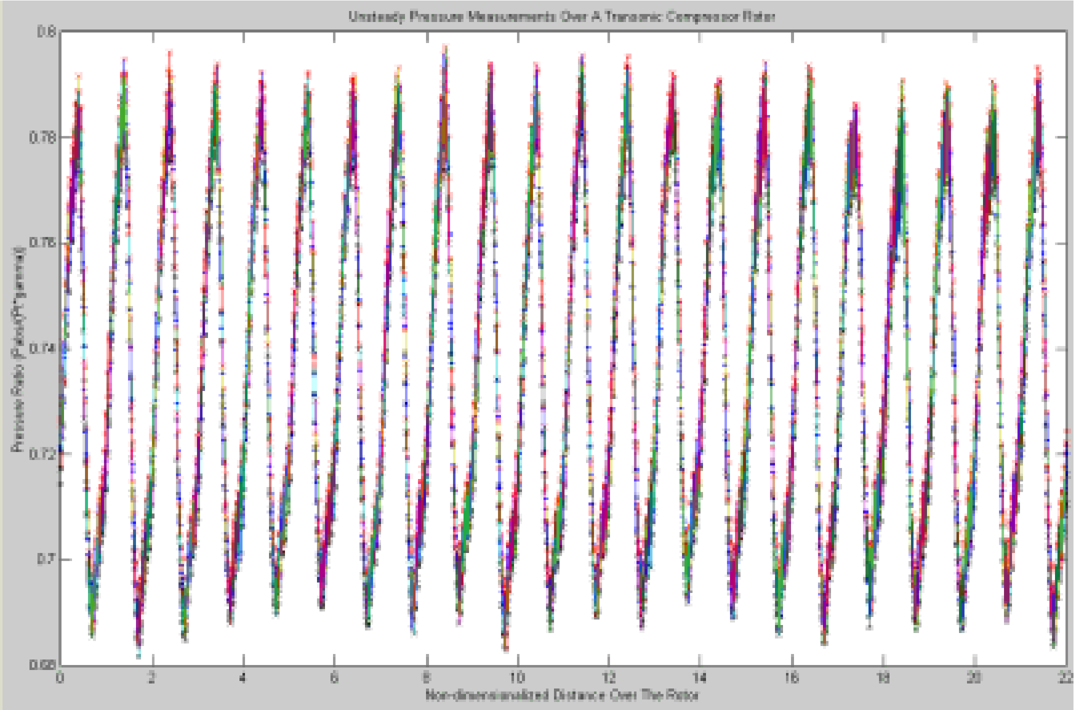


Figure 34. Unsteady Pressure over Rotor at 1.22 Pressure Ratio: Standard Deviation

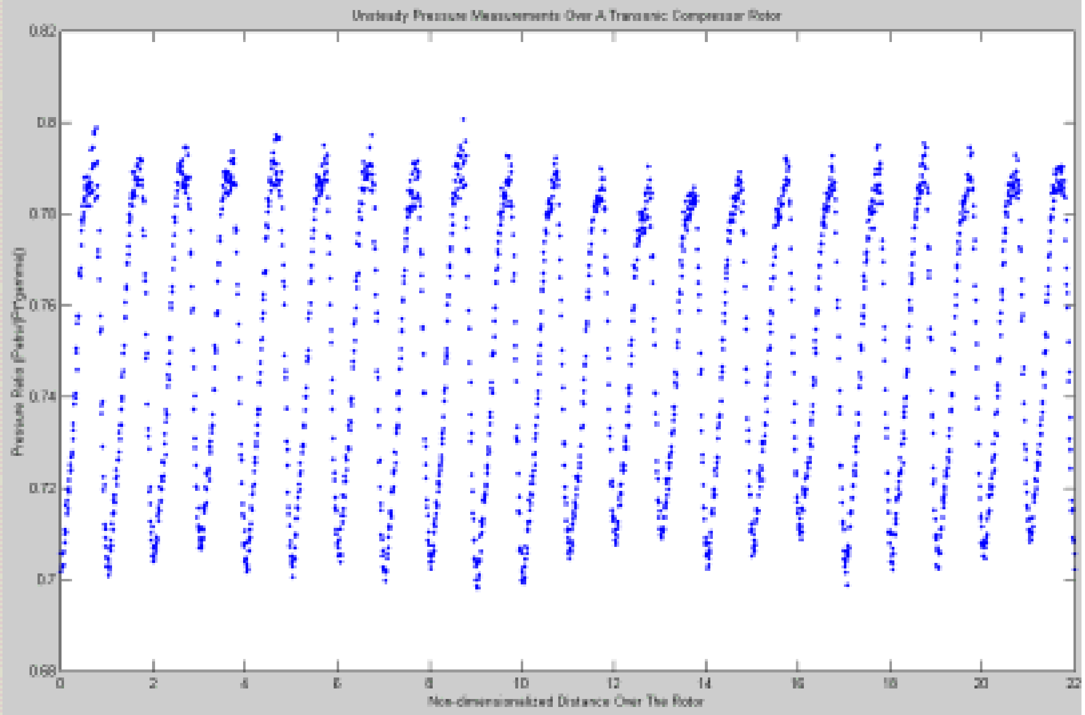


Figure 35. Unsteady Pressure over Rotor at 1.24 Pressure Ratio

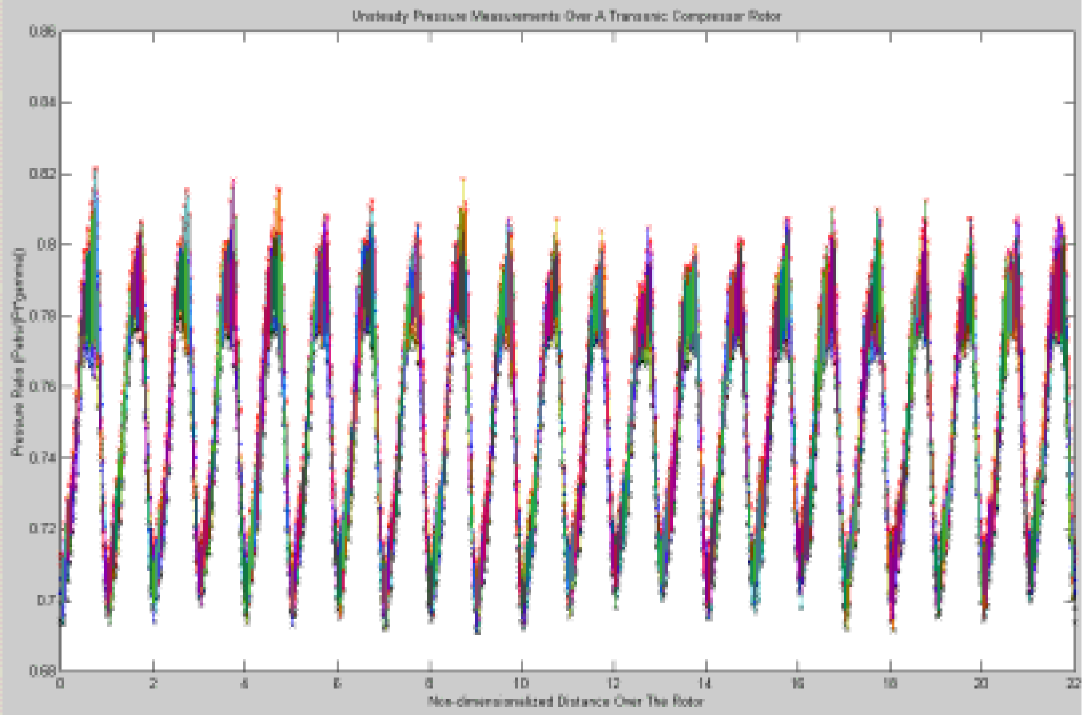


Figure 36. Unsteady Pressure over Rotor at 1.24 Pressure Ratio: Standard Deviation

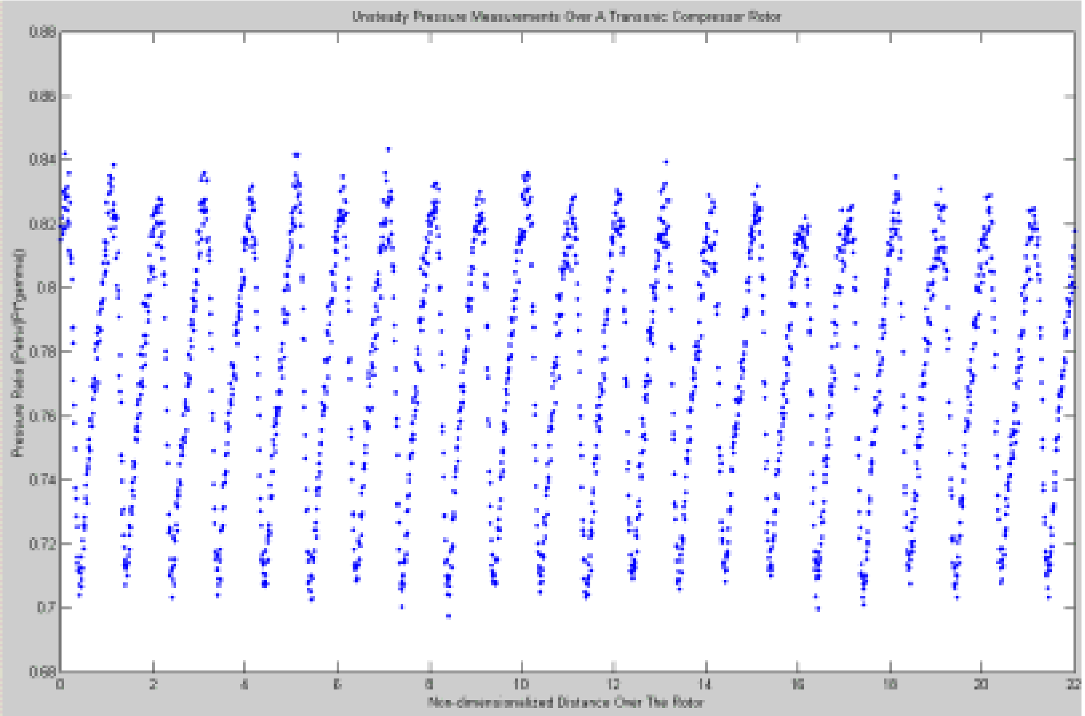


Figure 37. Unsteady Pressure over Rotor at 1.25 Pressure Ratio

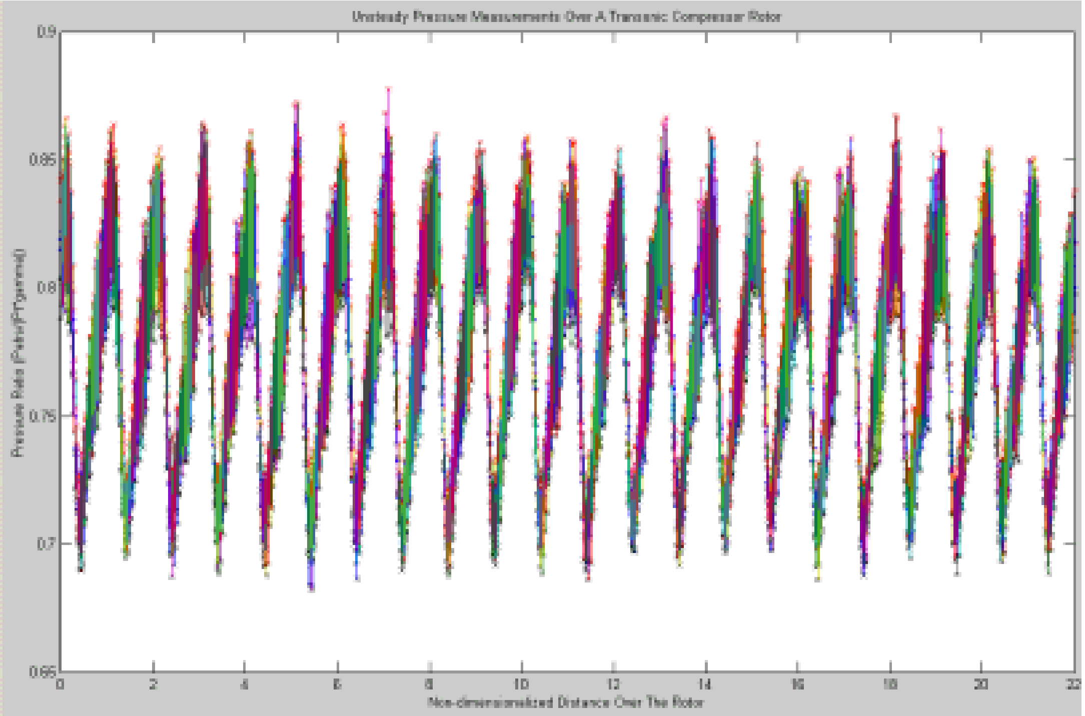


Figure 38. Unsteady Pressure over Rotor at 1.25 Pressure Ratio: Standard Deviation

C. 80% DESIGN SPEED 38% AXIAL CHORD

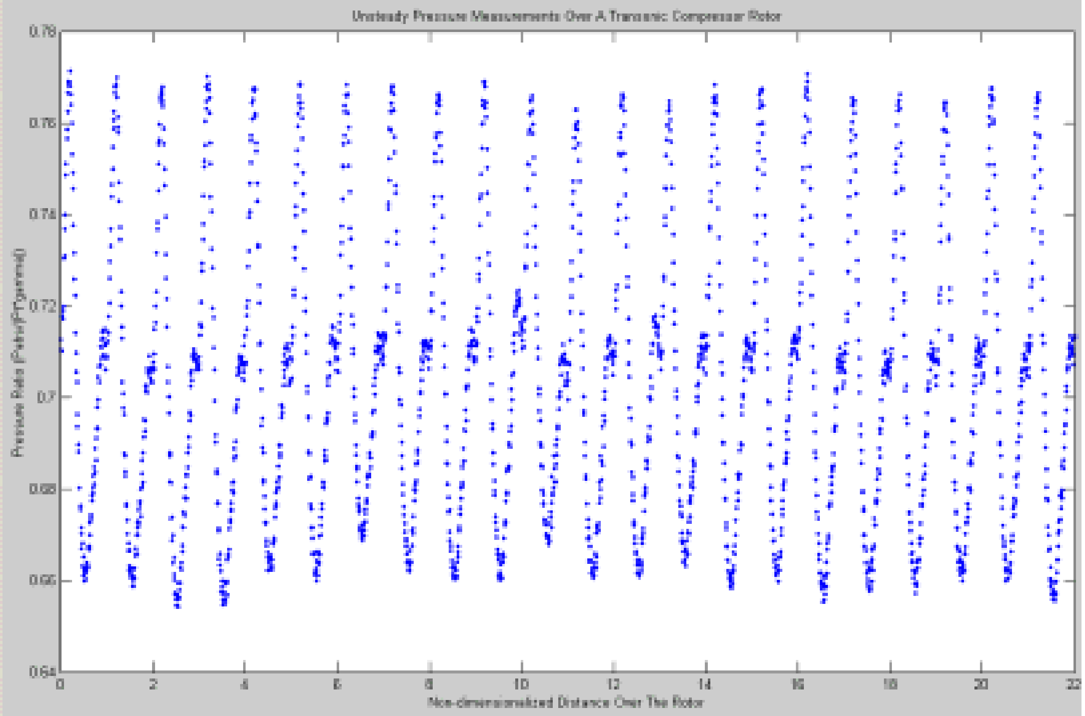


Figure 39. Unsteady Pressure over Rotor at 1.24 Pressure Ratio

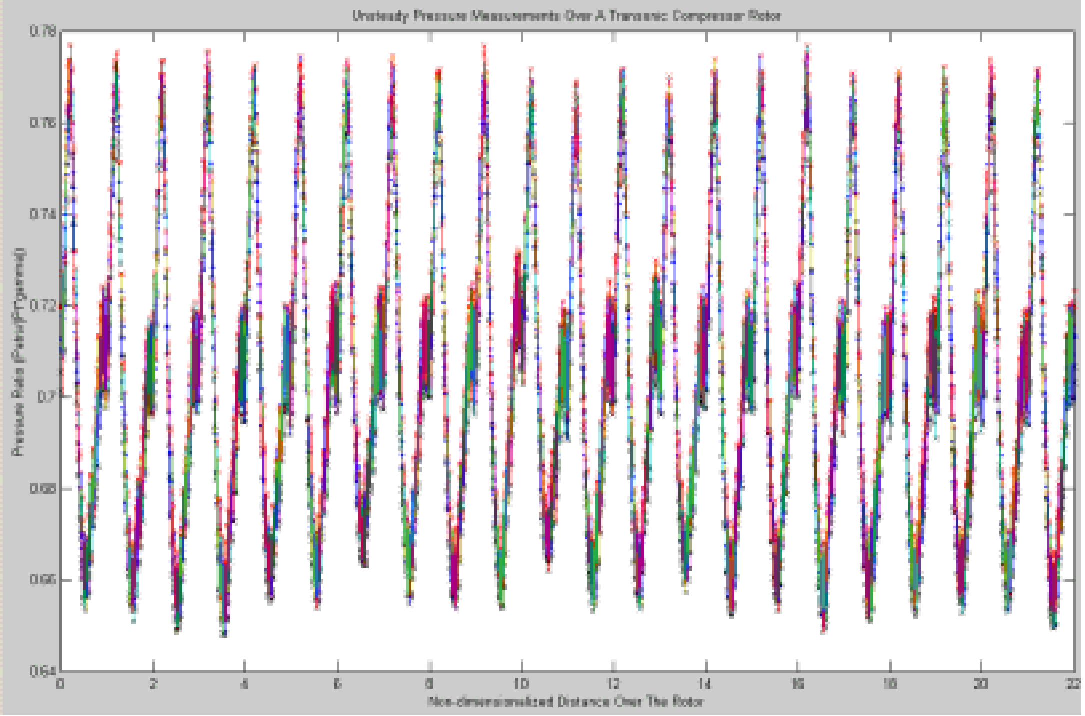


Figure 40. Unsteady Pressure over Rotor at 1.24 Pressure Ratio: Standard Deviation

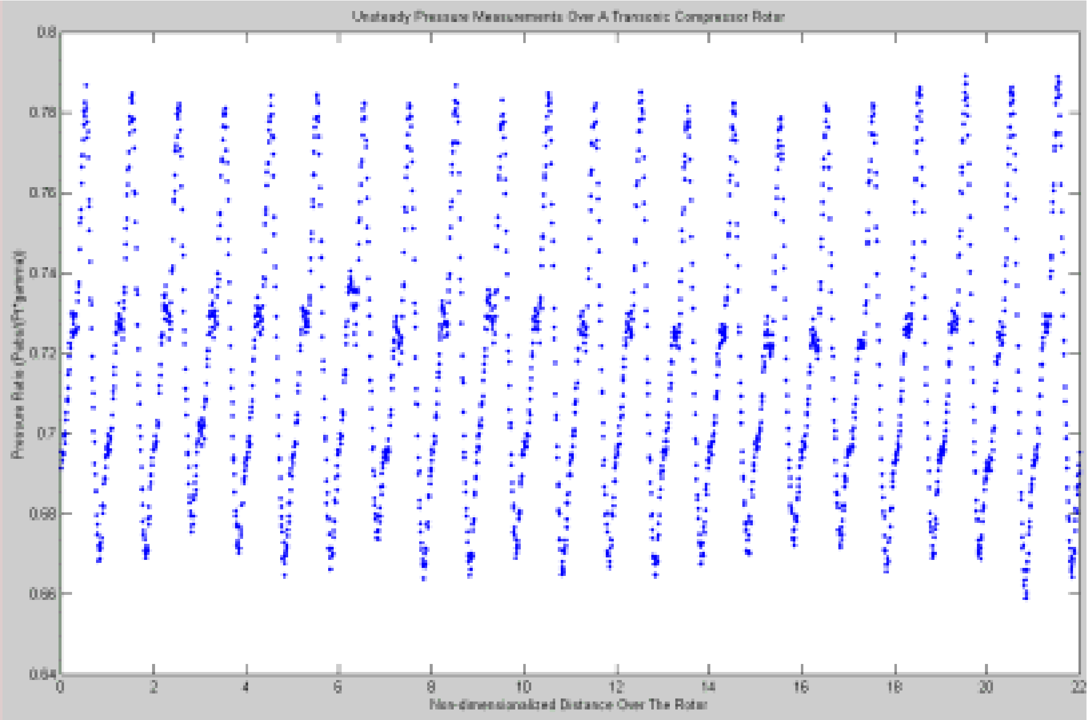


Figure 41. Unsteady Pressure over Rotor at 1.26 Pressure Ratio

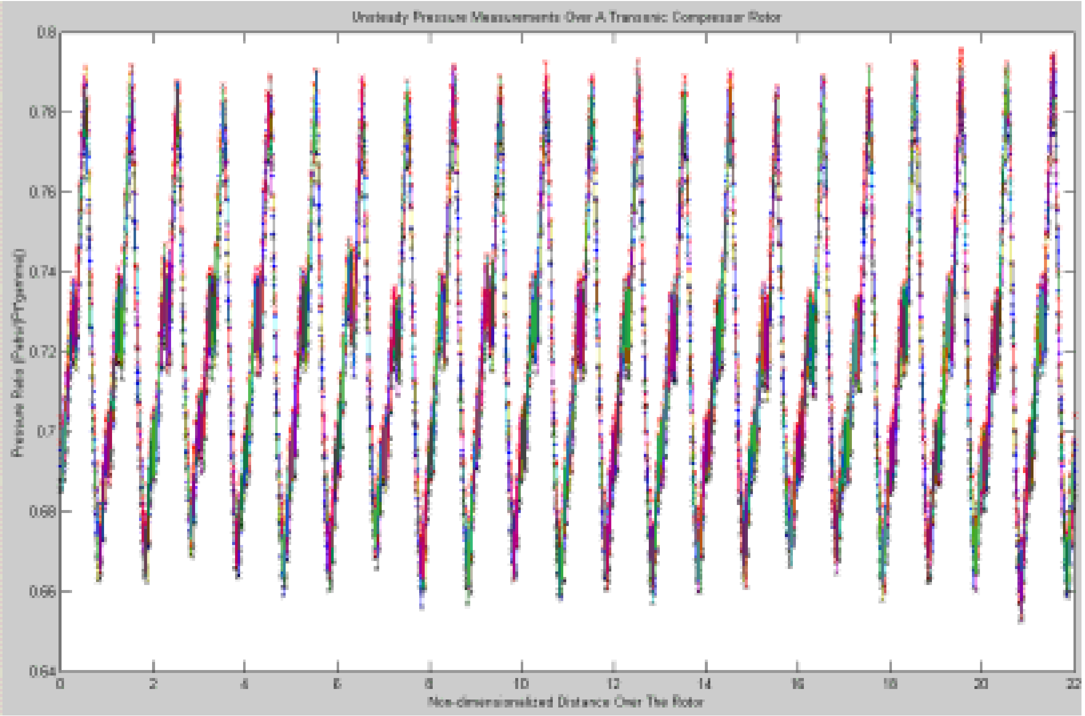


Figure 42. Unsteady Pressure over Rotor at 1.26 Pressure Ratio: Standard Deviation

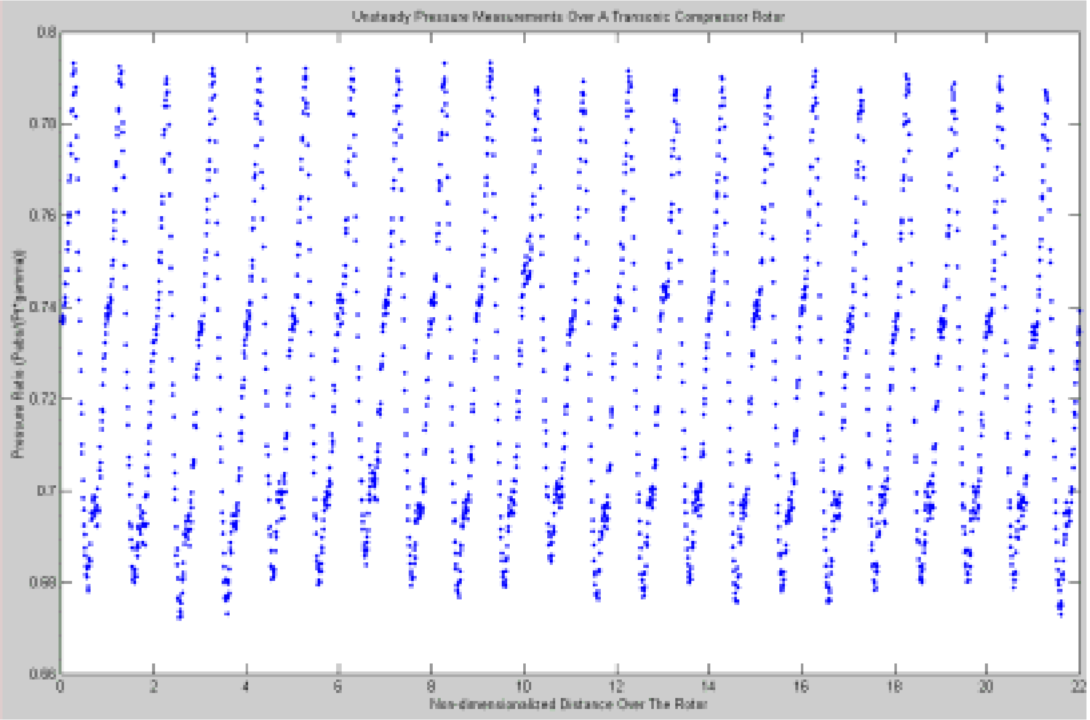


Figure 43. Unsteady Pressure over Rotor at 1.27 Pressure Ratio

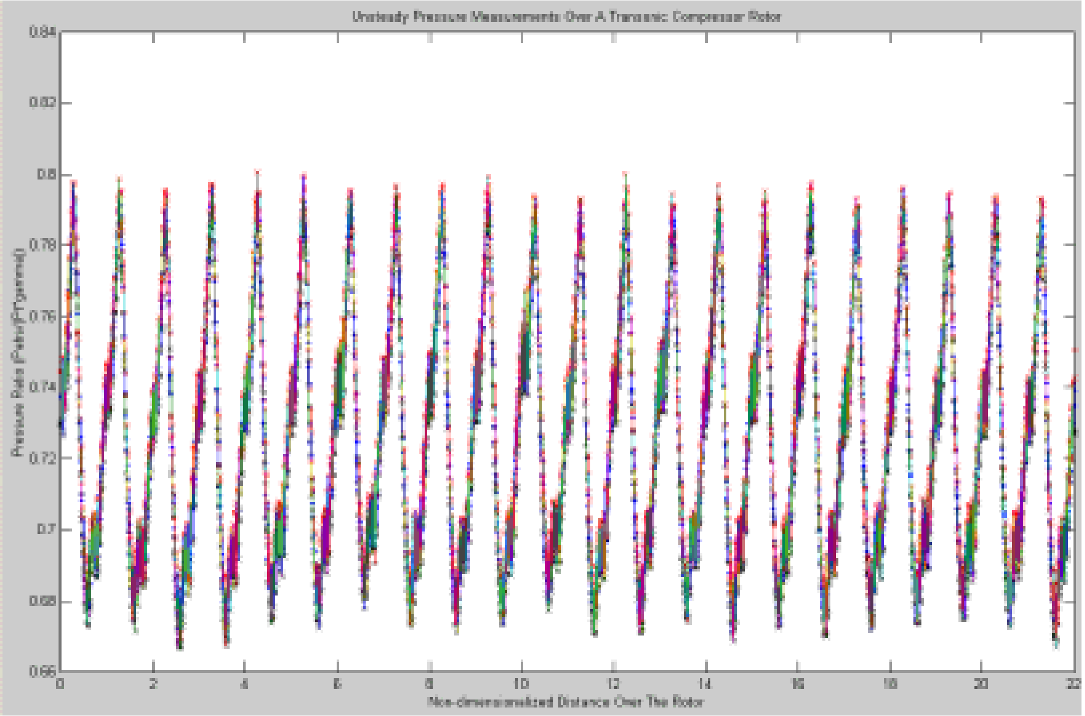


Figure 44. Unsteady Pressure over Rotor at 1.27 Pressure Ratio: Standard Deviation

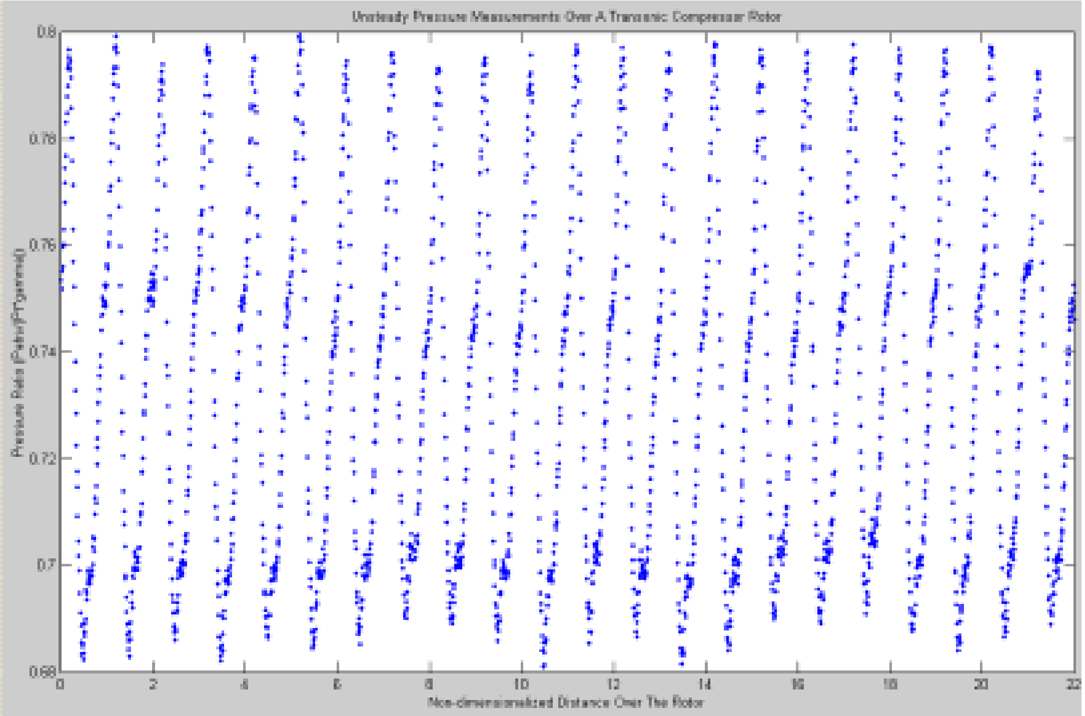


Figure 45. Unsteady Pressure over Rotor at 1.28 Pressure Ratio

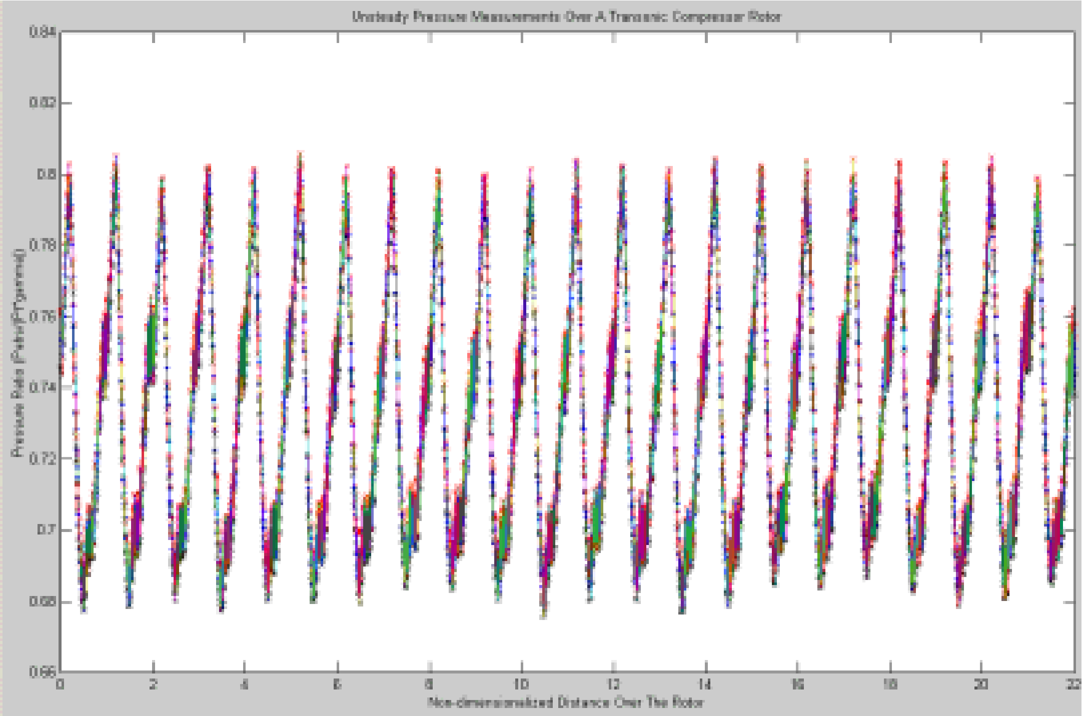


Figure 46. Unsteady Pressure over Rotor at 1.28 Pressure Ratio: Standard Deviation

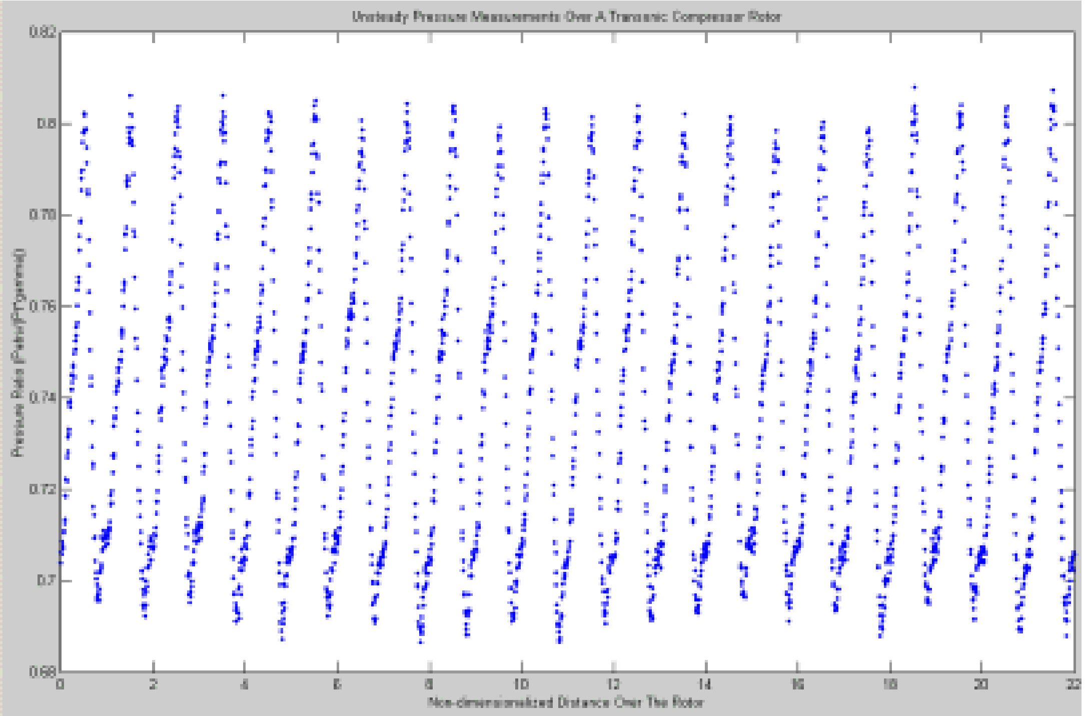


Figure 47. Unsteady Pressure over Rotor at 1.29 Pressure Ratio

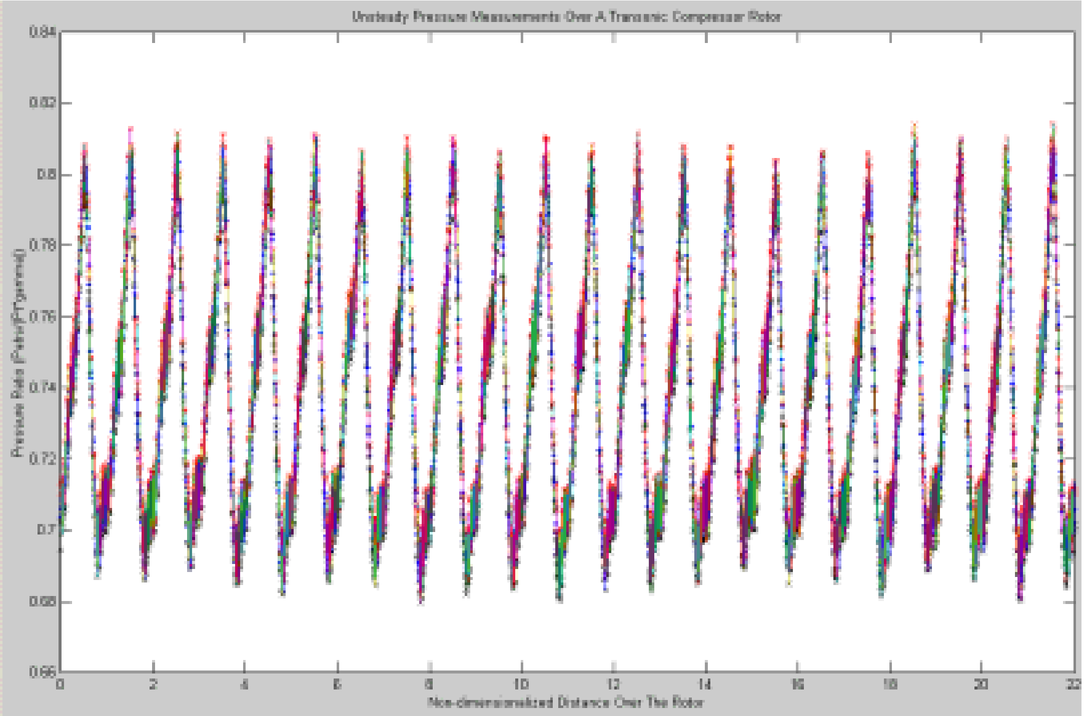


Figure 48. Unsteady Pressure over Rotor at 1.29 Pressure Ratio: Standard Deviation

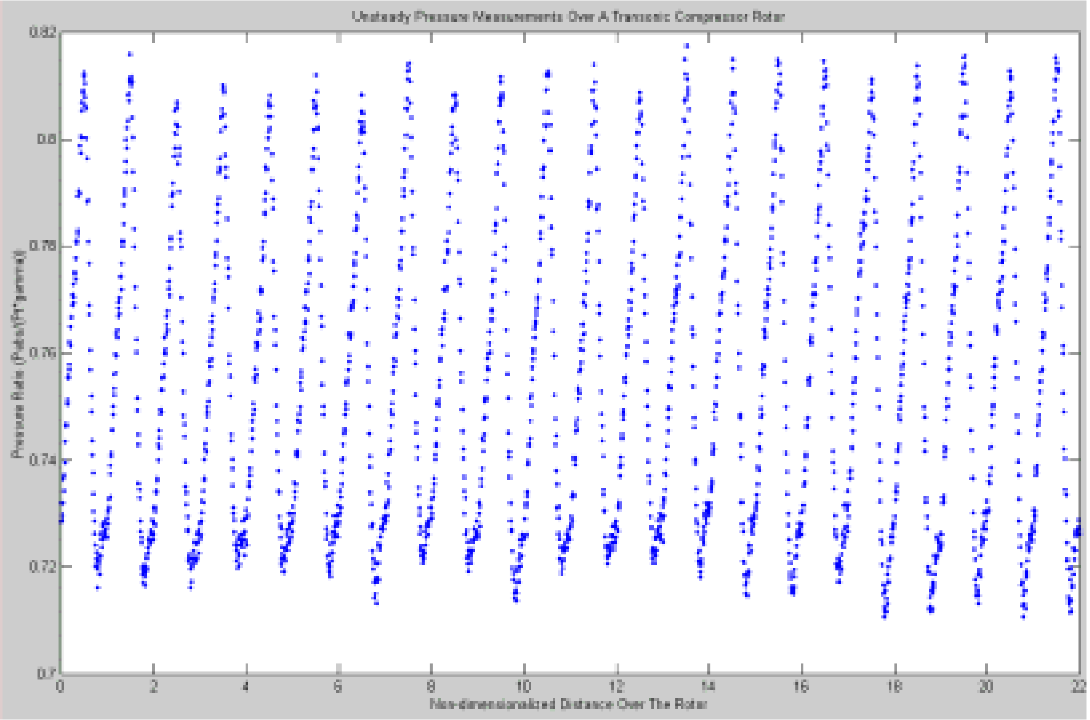


Figure 49. Unsteady Pressure over Rotor at 1.30 Pressure Ratio

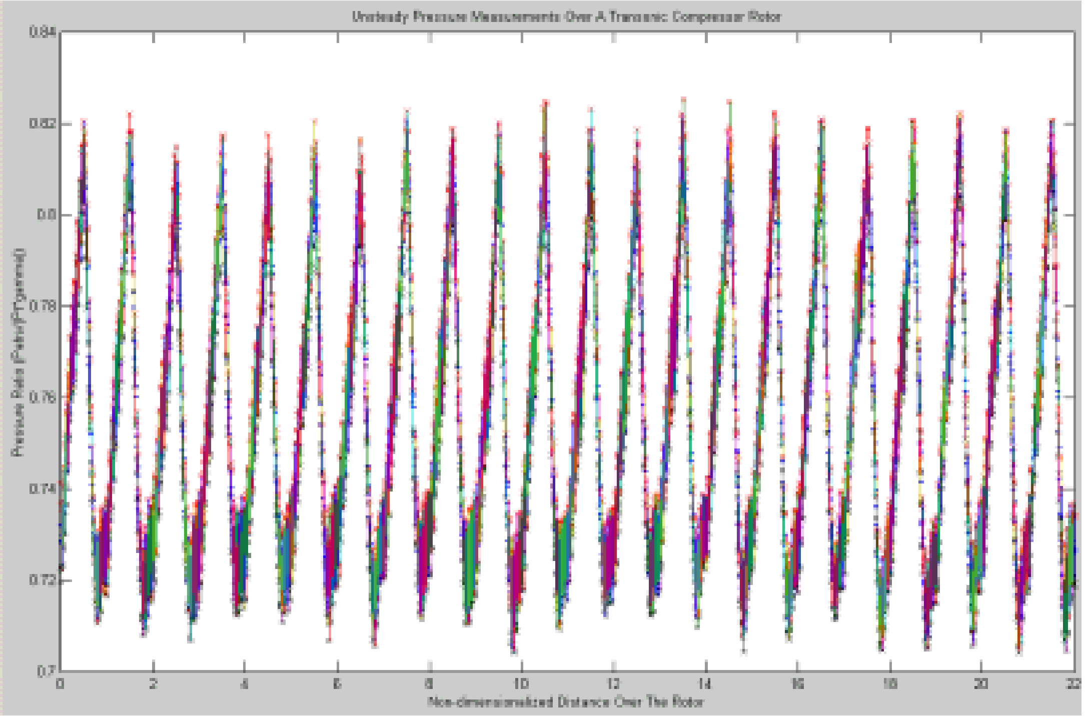


Figure 50. Unsteady Pressure over Rotor at 1.30 Pressure Ratio: Standard Deviation

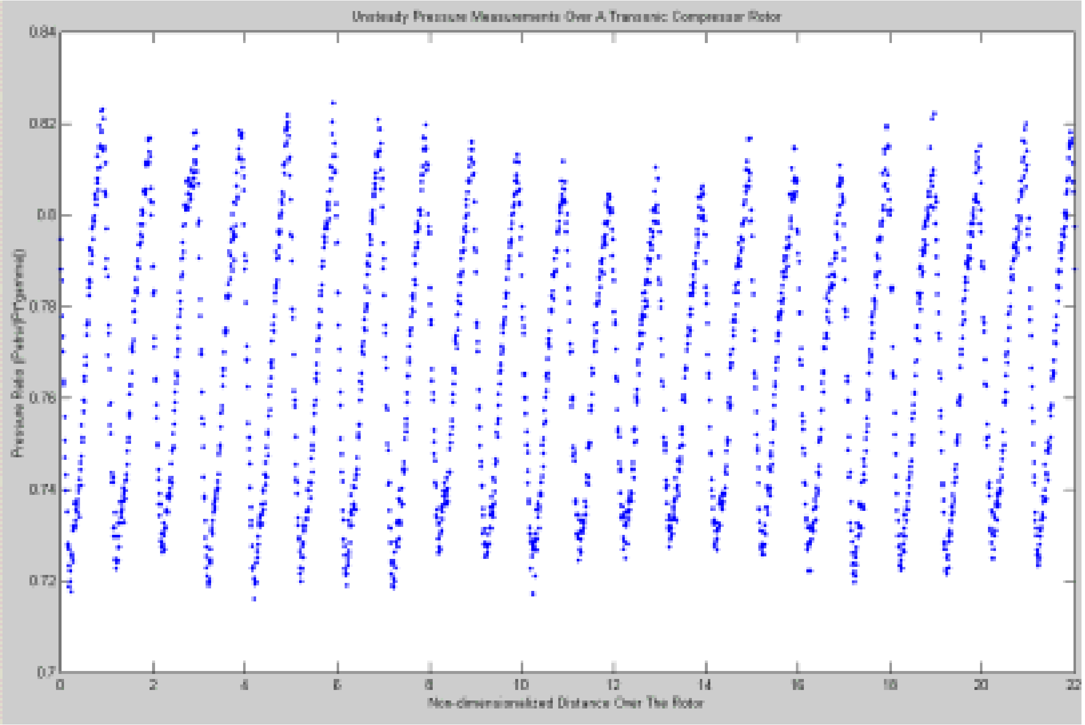


Figure 51. Unsteady Pressure over Rotor at 1.31 Pressure Ratio

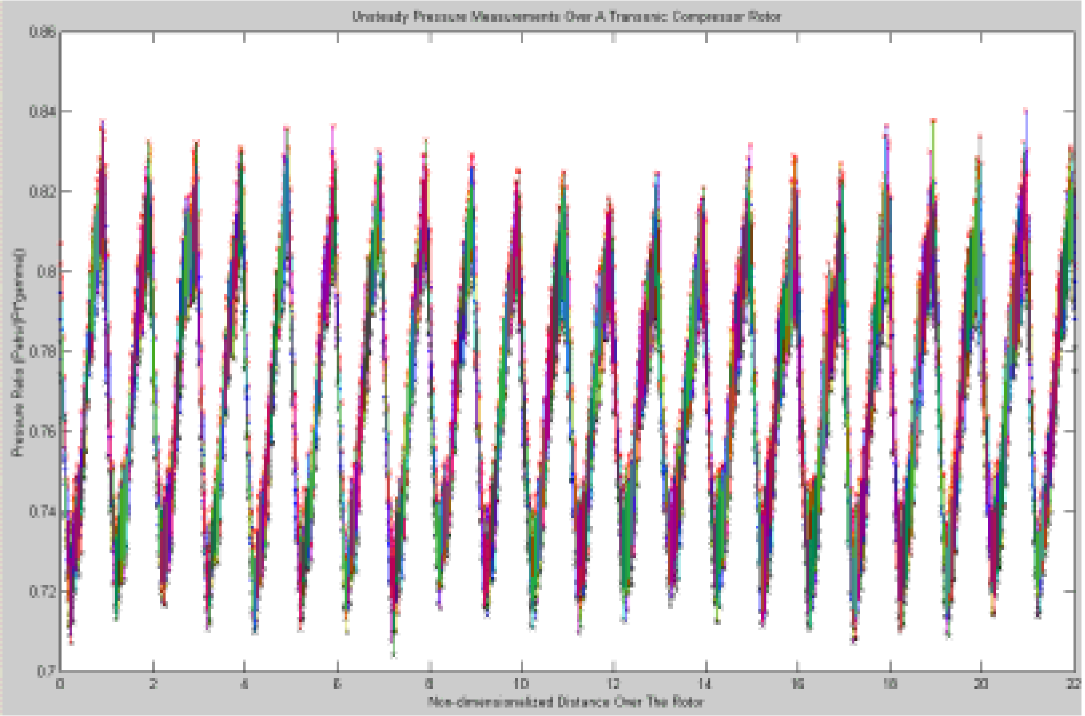


Figure 52. Unsteady Pressure over Rotor at 1.31 Pressure Ratio: Standard Deviation

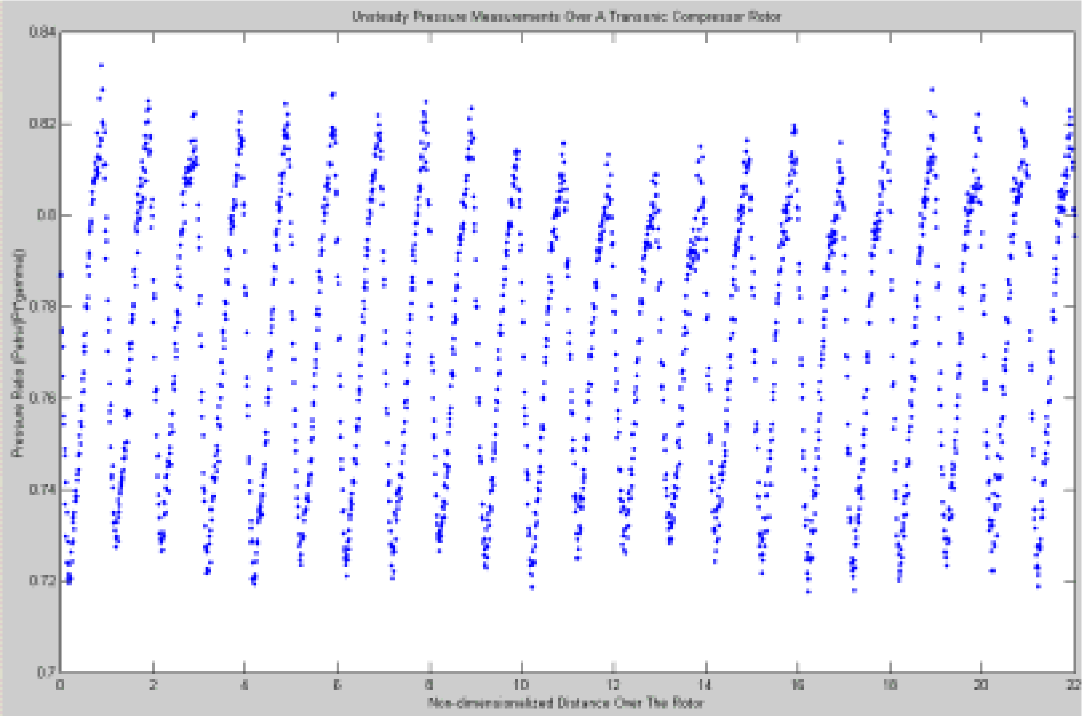


Figure 53. Unsteady Pressure over Rotor at 1.315 Pressure Ratio

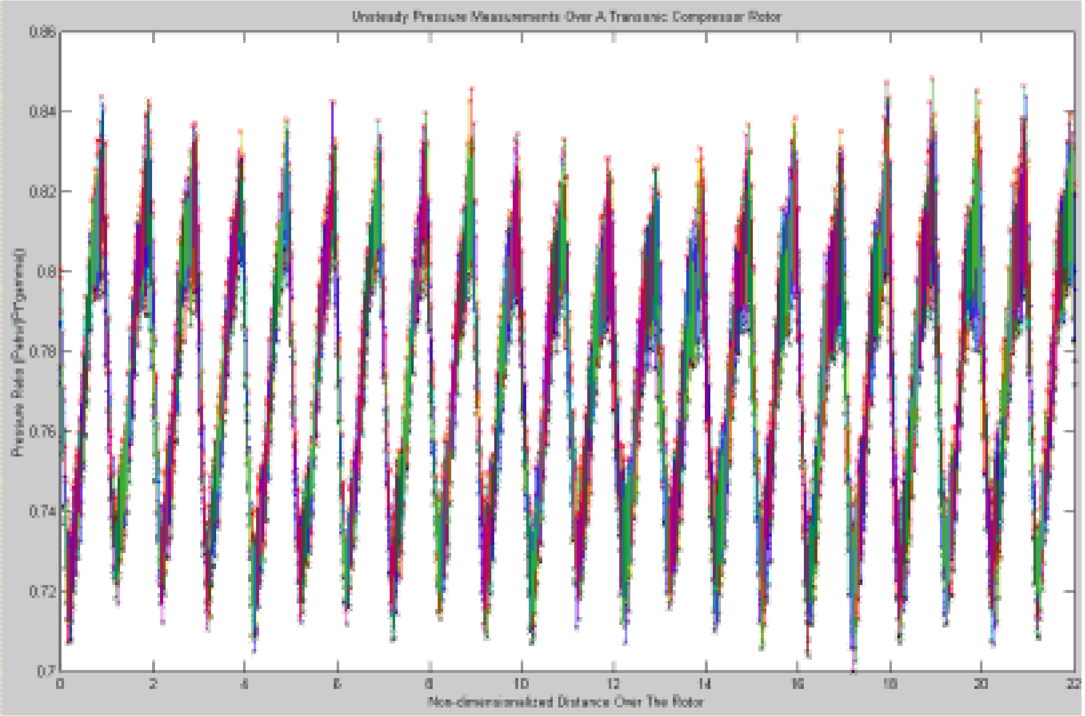


Figure 54. Unsteady Pressure over Rotor at 1.315 Pressure Ratio: Standard Deviation

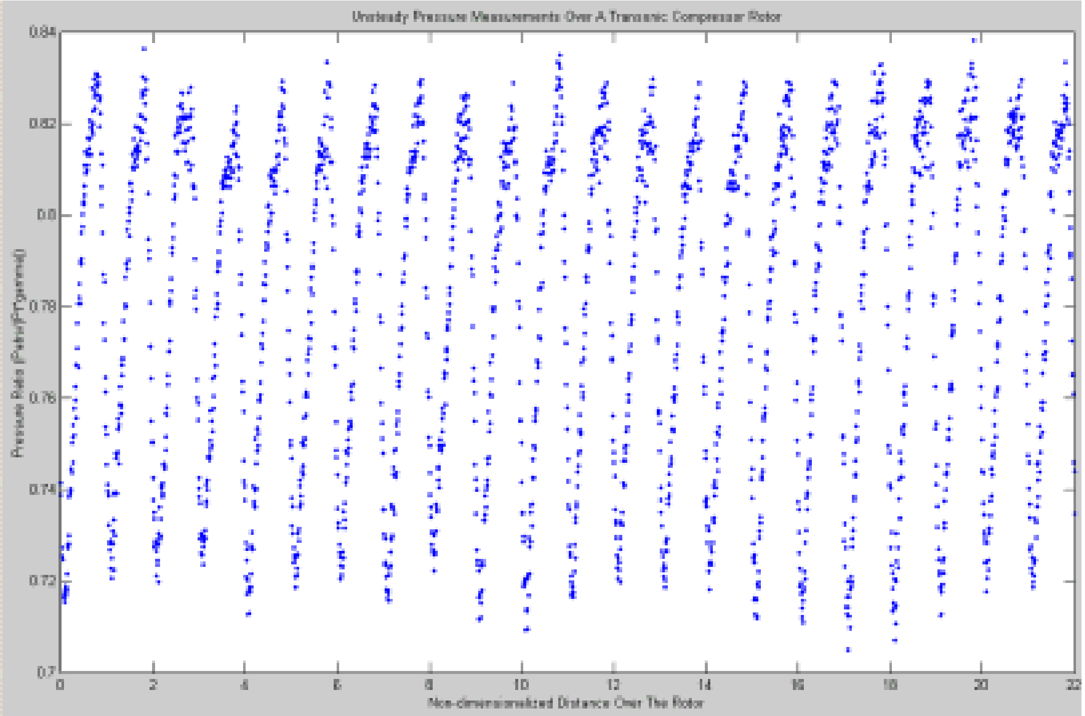


Figure 55. Unsteady Pressure over Rotor at 1.32 Pressure Ratio

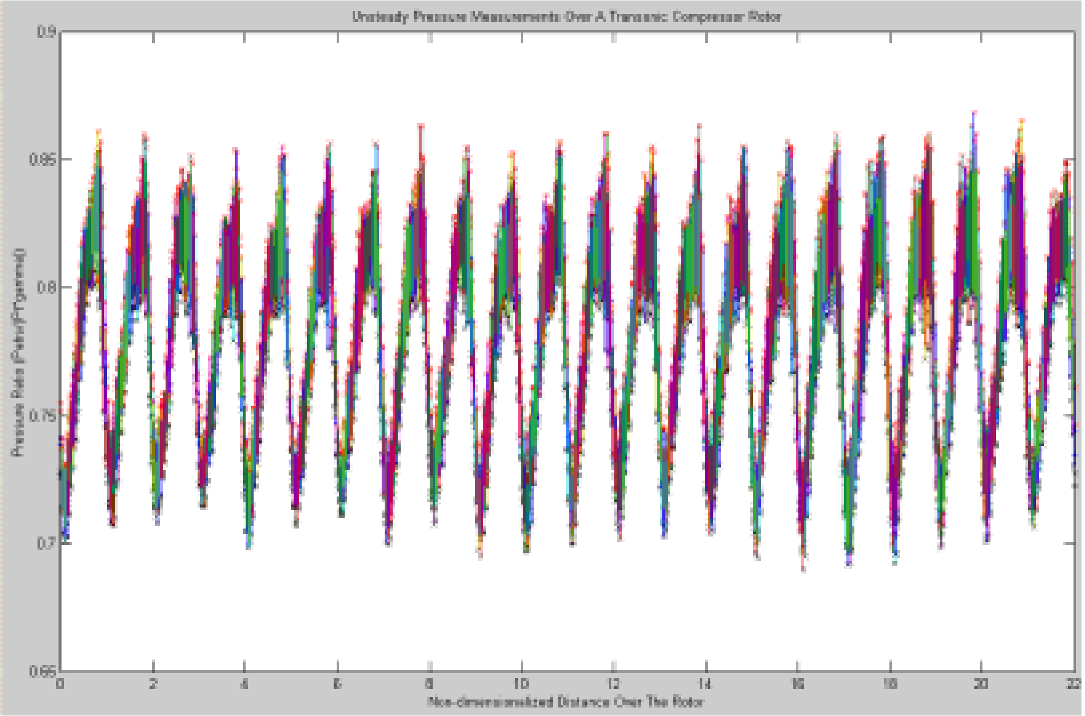


Figure 56. Unsteady Pressure over Rotor at 1.32 Pressure Ratio: Standard Deviation

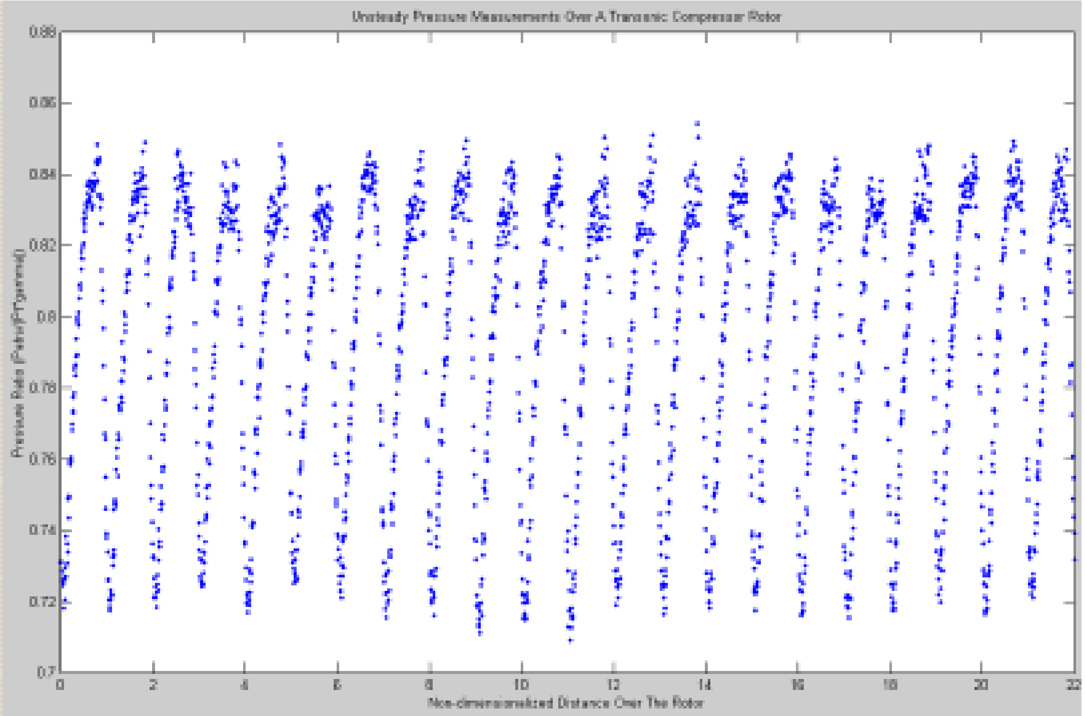


Figure 57. Unsteady Pressure over Rotor at 1.33 Pressure Ratio

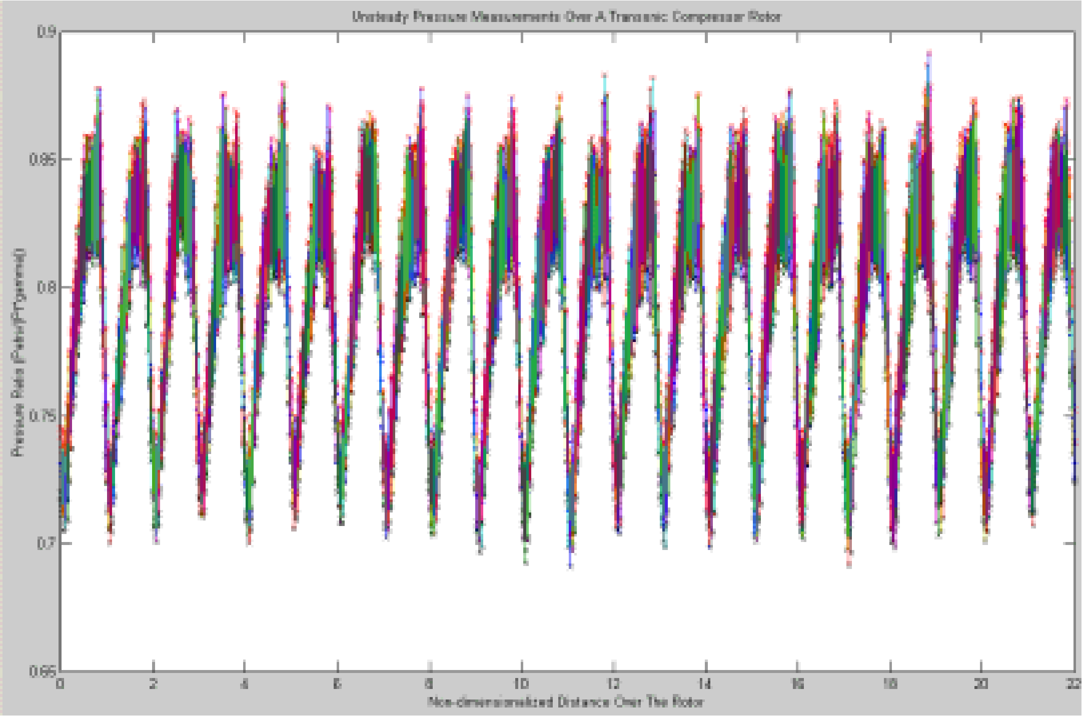


Figure 58. Unsteady Pressure over Rotor at 1.33 Pressure Ratio: Standard Deviation

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B: COMPARISON OF EXPERIMENTAL AND COMPUTATIONAL DATA

A. 60% DESIGN SPEED 34% AXIAL CHORD

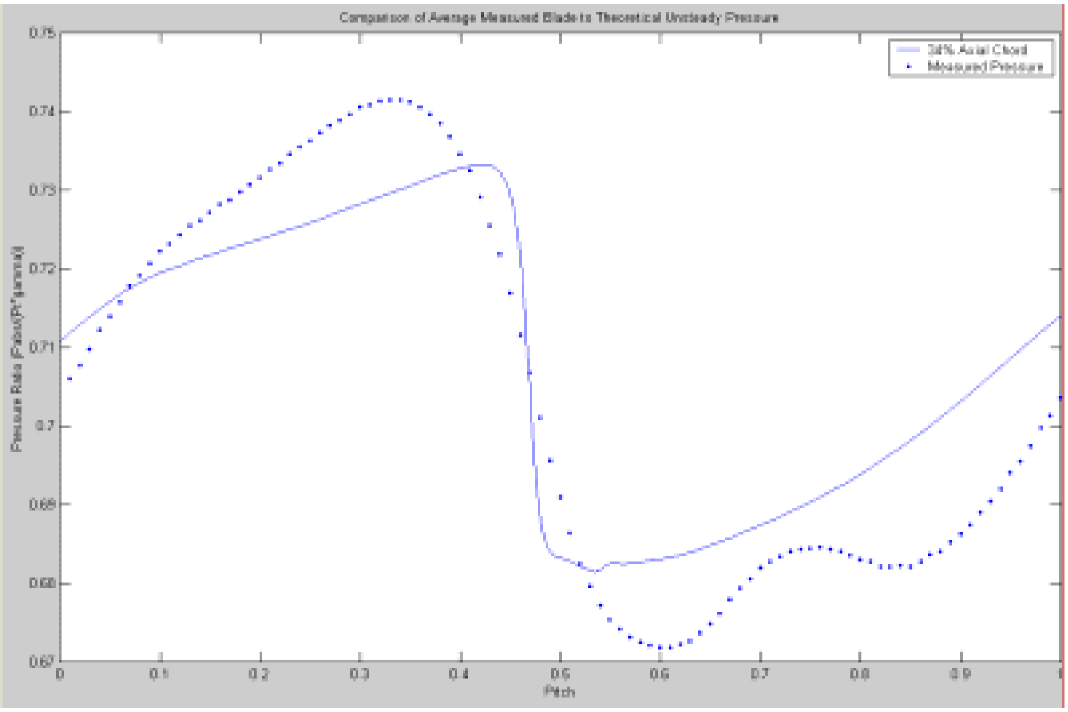


Figure 59. Unsteady Pressure at 1.12 Pressure Ratio

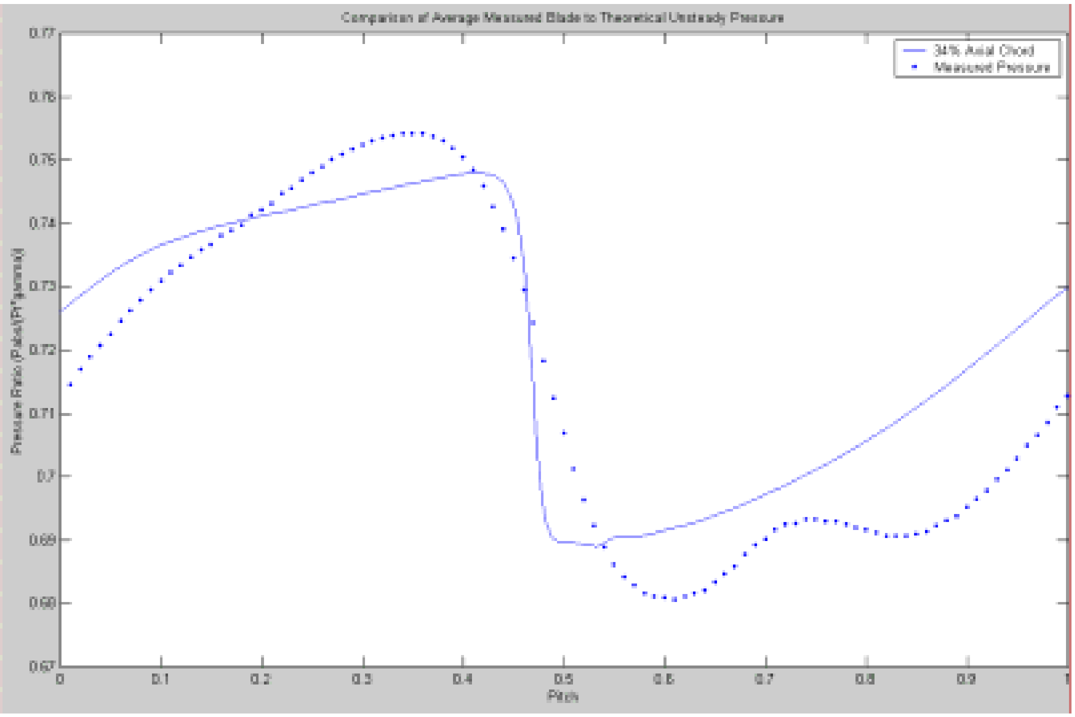


Figure 60. Unsteady Pressure at 1.13 Pressure Ratio

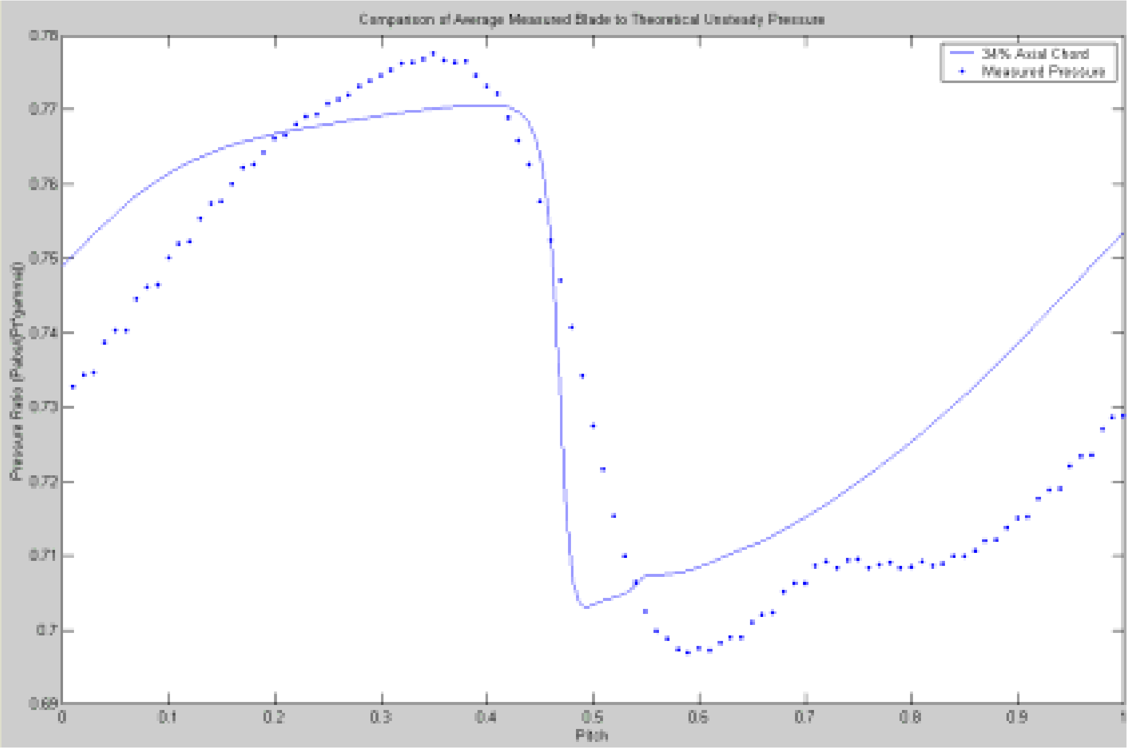


Figure 61. Unsteady Pressure at 1.15 Pressure Ratio

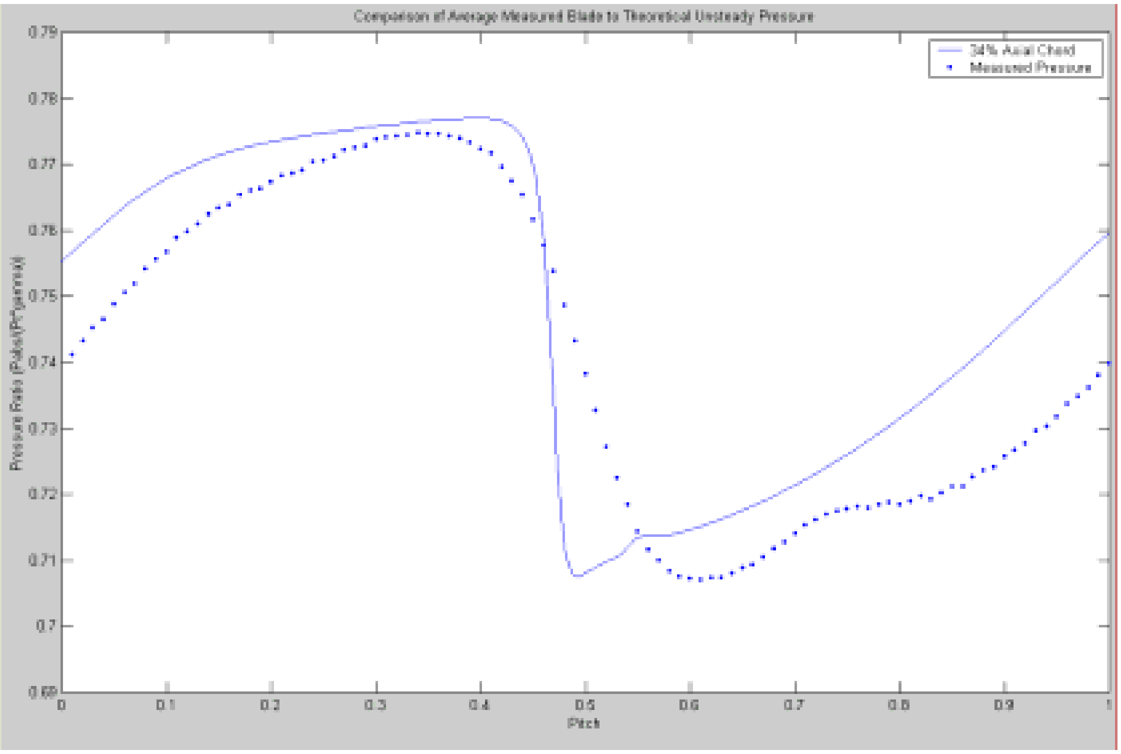


Figure 62. Unsteady Pressure at 1.16 Pressure Ratio

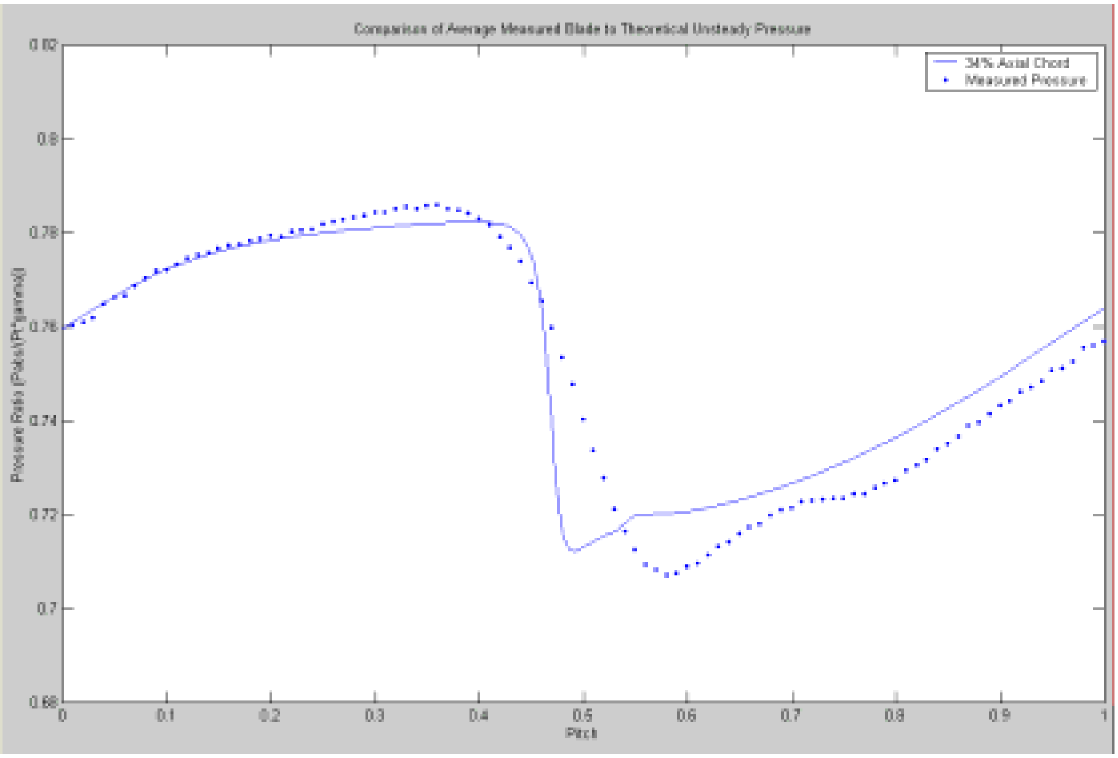


Figure 63. Unsteady Pressure at 1.18 Pressure Ratio

Table 5 Computationally Predicted Unsteady Pressure Data at 34% Chord, 60% Design Speed

Pitch	PR = 1.12	PR = 1.13	PR = 1.15	PR = 1.16	PR = 1.18
0.000000	0.710804	0.726059	0.748992	0.755186	0.759613
0.015290	0.712472	0.728056	0.751269	0.757462	0.761818
0.031117	0.714073	0.729974	0.753490	0.759677	0.763965
0.048311	0.715726	0.731971	0.755818	0.762017	0.766245
0.066147	0.717281	0.733870	0.758084	0.764322	0.768536
0.084093	0.718561	0.735459	0.760052	0.766355	0.770619
0.101897	0.719564	0.736719	0.761662	0.768045	0.772413
0.119562	0.720434	0.737769	0.762991	0.769447	0.773940
0.137093	0.721238	0.738687	0.764112	0.770628	0.775248
0.154525	0.721978	0.739478	0.765034	0.771600	0.776346
0.171892	0.722673	0.740167	0.765792	0.772394	0.777263
0.189220	0.723356	0.740794	0.766428	0.773054	0.778034
0.206526	0.724041	0.741381	0.766970	0.773608	0.778687
0.223822	0.724754	0.741961	0.767453	0.774090	0.779249
0.241106	0.725518	0.742566	0.767911	0.774535	0.779751
0.258352	0.726296	0.743167	0.768330	0.774934	0.780197
0.275548	0.727088	0.743768	0.768723	0.775301	0.780602
0.292678	0.727888	0.744366	0.769091	0.775639	0.780967
0.309717	0.728692	0.744961	0.769437	0.775951	0.781301
0.326625	0.729498	0.745551	0.769767	0.776241	0.781607
0.343344	0.730296	0.746126	0.770071	0.776502	0.781880
0.359775	0.731073	0.746675	0.770344	0.776728	0.782115
0.375761	0.731806	0.747177	0.770566	0.776901	0.782295
0.391067	0.732462	0.747598	0.770703	0.776991	0.782389
0.405368	0.732986	0.747881	0.770700	0.776941	0.782342
0.418293	0.733286	0.747932	0.770457	0.776651	0.782049
0.429520	0.733197	0.747566	0.769771	0.775913	0.781300
0.438880	0.732518	0.746570	0.768412	0.774489	0.779853
0.446402	0.730939	0.744586	0.765971	0.771961	0.777284
0.452279	0.728452	0.741623	0.762467	0.768344	0.773607
0.456774	0.724965	0.737535	0.757697	0.763429	0.768611
0.460171	0.721233	0.733215	0.752688	0.758268	0.763356
0.462717	0.717265	0.728598	0.747324	0.752744	0.757754
0.464616	0.714145	0.724998	0.743156	0.748452	0.753394
0.466029	0.711202	0.721550	0.739128	0.744304	0.749207
0.467078	0.709569	0.719663	0.736940	0.742048	0.746910
0.467856	0.707707	0.717463	0.734359	0.739391	0.744240
0.468433	0.707313	0.716986	0.733779	0.738783	0.743603
0.468860	0.705948	0.715384	0.731922	0.736882	0.741711
0.469177	0.706307	0.715757	0.732302	0.737255	0.742058
0.469412	0.705077	0.714332	0.730691	0.735619	0.740450
0.470418	0.701556	0.710340	0.726111	0.730906	0.735663
0.472836	0.697110	0.705102	0.719919	0.724490	0.729092
0.476488	0.691422	0.698466	0.712193	0.716529	0.721013
0.481207	0.686954	0.693354	0.706409	0.710638	0.715111
0.486836	0.684520	0.690698	0.703659	0.707944	0.712523
0.493197	0.683492	0.689737	0.703005	0.707461	0.712233

Table 4 Continued

Pitch	PR = 1.12	PR = 1.13	PR = 1.15	PR = 1.16	PR = 1.18
0.500080	0.683213	0.689669	0.703449	0.708138	0.713146
0.507240	0.682930	0.689615	0.703934	0.708869	0.714125
0.514387	0.682501	0.689404	0.704228	0.709406	0.714919
0.521135	0.682117	0.689240	0.704510	0.709897	0.715596
0.527131	0.681772	0.689094	0.704750	0.710319	0.716168
0.531991	0.681526	0.689017	0.704992	0.710715	0.716694
0.535304	0.681477	0.689096	0.705296	0.711127	0.717203
0.536668	0.681597	0.689299	0.705616	0.711495	0.717615
0.536932	0.681665	0.689394	0.705743	0.711635	0.717757
0.537289	0.681616	0.689332	0.705674	0.711573	0.717705
0.537770	0.681663	0.689399	0.705774	0.711689	0.717832
0.538420	0.681670	0.689409	0.705801	0.711732	0.717891
0.539297	0.681731	0.689493	0.705931	0.711887	0.718069
0.540479	0.681819	0.689609	0.706098	0.712084	0.718294
0.542070	0.681961	0.689794	0.706361	0.712388	0.718636
0.544208	0.682161	0.690051	0.706709	0.712783	0.719075
0.547071	0.682380	0.690326	0.707084	0.713208	0.719545
0.550885	0.682548	0.690532	0.707360	0.713527	0.719904
0.555919	0.682580	0.690568	0.707455	0.713663	0.720073
0.562468	0.682524	0.690514	0.707454	0.713685	0.720105
0.570801	0.682538	0.690509	0.707509	0.713742	0.720126
0.580999	0.682696	0.690743	0.707712	0.713918	0.720172
0.593088	0.683026	0.691263	0.708159	0.714330	0.720424
0.606836	0.683213	0.691921	0.708846	0.714980	0.720917
0.621848	0.683625	0.692447	0.709709	0.715829	0.721618
0.637714	0.684343	0.693134	0.710683	0.716825	0.722482
0.654071	0.685074	0.694167	0.711681	0.717898	0.723436
0.670666	0.685859	0.695232	0.712793	0.719037	0.724454
0.687414	0.686704	0.696358	0.714111	0.720338	0.725623
0.704233	0.687603	0.697553	0.715531	0.721756	0.726960
0.721079	0.688555	0.698809	0.717044	0.723275	0.728418
0.737926	0.689566	0.700127	0.718652	0.724891	0.729981
0.754757	0.690644	0.701517	0.720352	0.726596	0.731643
0.771536	0.691805	0.702988	0.722137	0.728378	0.733387
0.788280	0.693039	0.704537	0.724004	0.730241	0.735214
0.804995	0.694347	0.706163	0.725952	0.732187	0.737123
0.821678	0.695728	0.707867	0.727979	0.734212	0.739108
0.838330	0.697181	0.709650	0.730085	0.736314	0.741167
0.854951	0.698707	0.711514	0.732269	0.738493	0.743301
0.871542	0.700302	0.713453	0.734527	0.740745	0.745504
0.888098	0.701965	0.715465	0.736854	0.743066	0.747775
0.904609	0.703690	0.717542	0.739247	0.745452	0.750108
0.921048	0.705463	0.719670	0.741688	0.747886	0.752488
0.937367	0.707258	0.721821	0.744153	0.750348	0.754893
0.953474	0.709047	0.723963	0.746609	0.752801	0.757286
0.969210	0.710804	0.726059	0.748992	0.755186	0.759613
0.984342	0.712471	0.728056	0.751269	0.757462	0.761818
1.000000	0.714073	0.729974	0.753490	0.759677	0.763965

B. 70% DESIGN SPEED 38% AXIAL CHORD

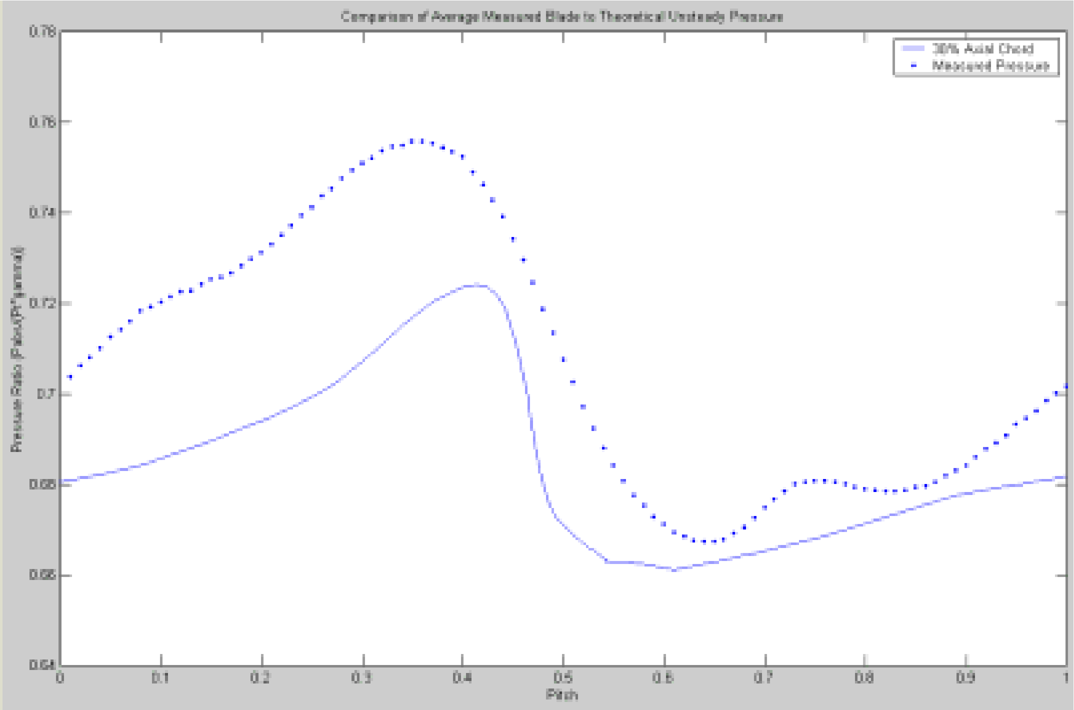


Figure 64. Unsteady Pressure at 1.18 Pressure Ratio

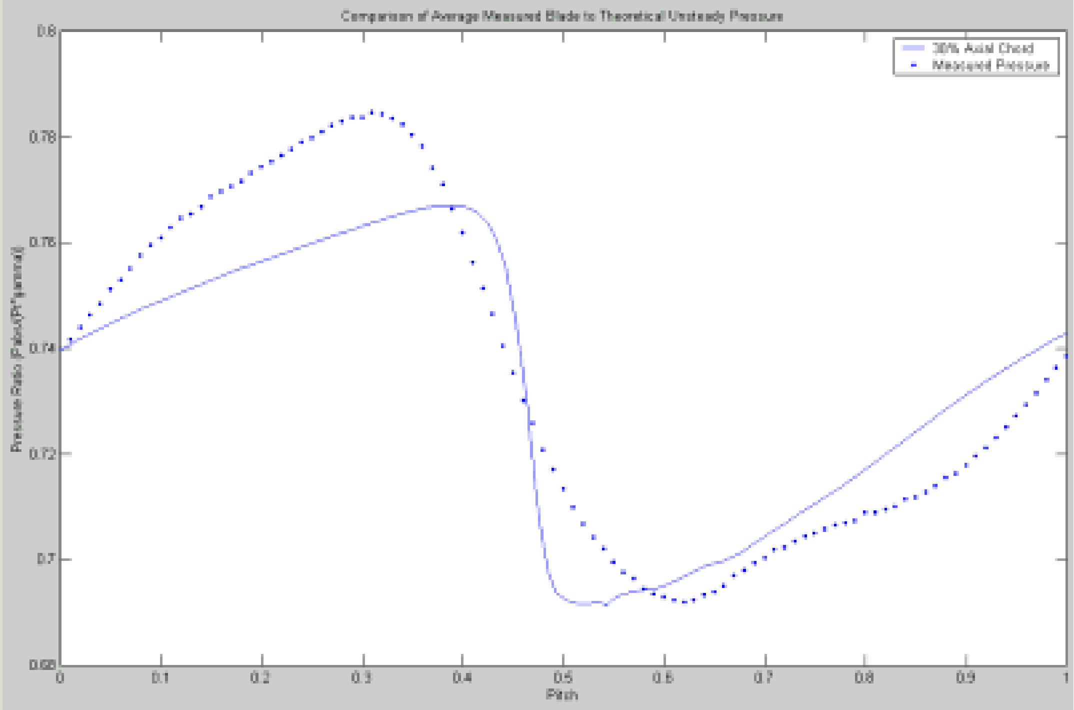


Figure 65. Unsteady Pressure at 1.22 Pressure Ratio

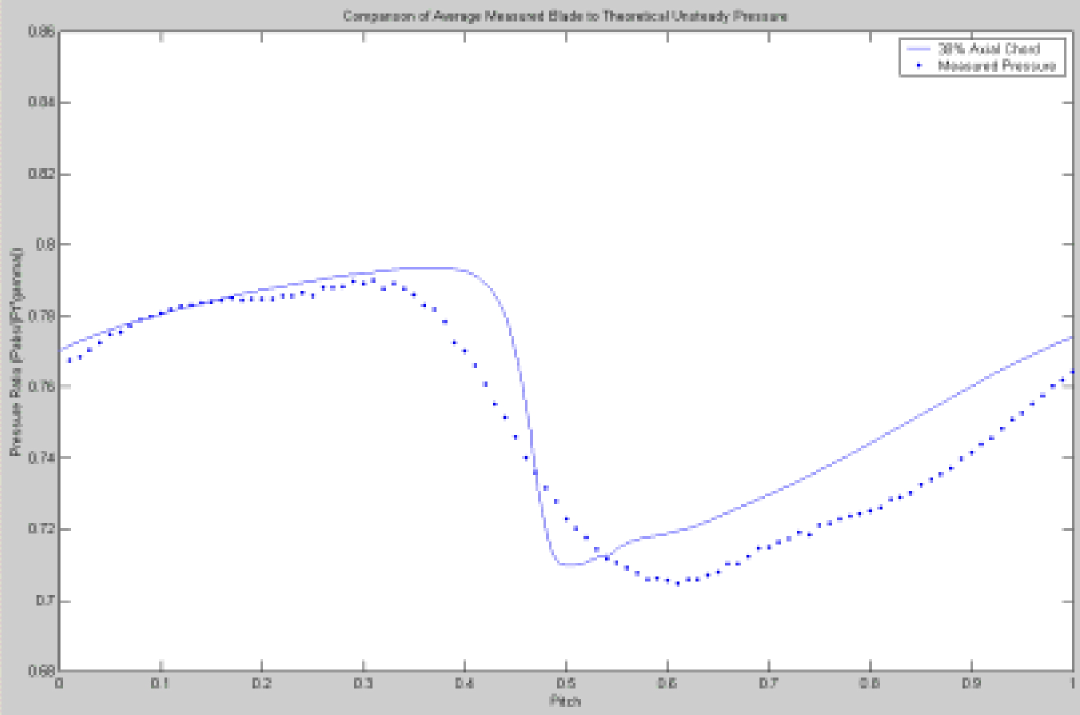


Figure 66. Unsteady Pressure at 1.24 Pressure Ratio

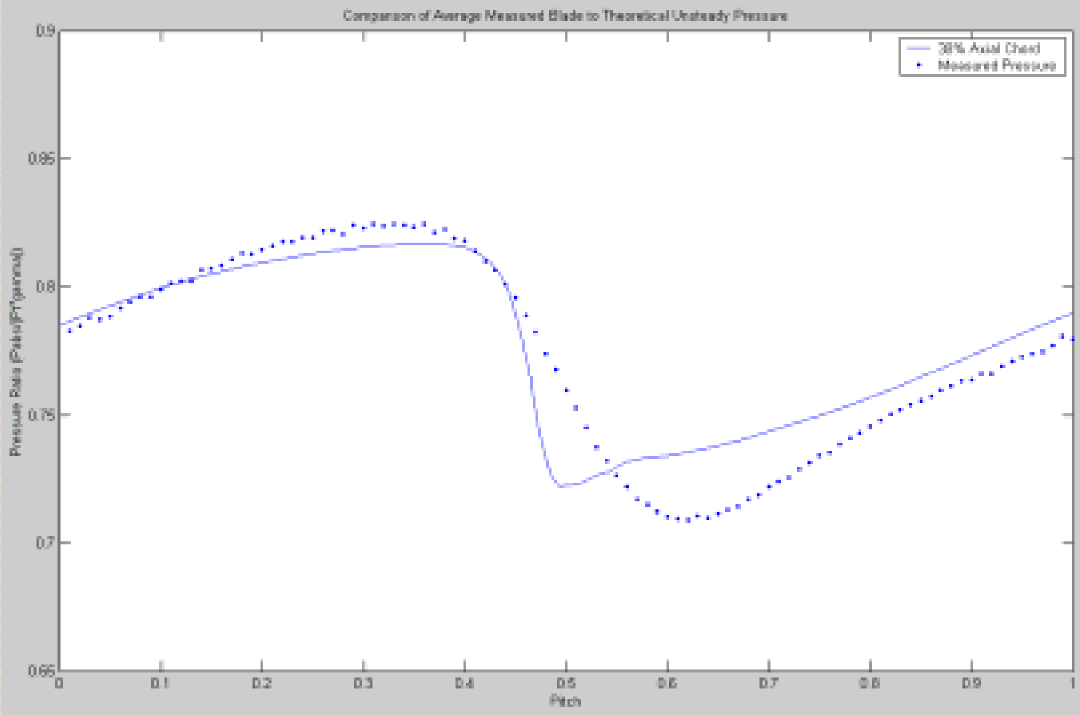


Figure 67. Unsteady Pressure at 1.25 Pressure Ratio

Table 6 Computationally Predicted Unsteady Pressure Data at 38% Chord, 70% Design Speed

Pitch	PR = 1.18	PR = 1.22	PR = 1.24	PR = 1.25
0.000000	0.680577	0.739680	0.770300	0.785184
0.015303	0.681075	0.741282	0.772194	0.787599
0.030883	0.681763	0.742892	0.774036	0.789999
0.047476	0.682503	0.744542	0.775897	0.792527
0.064626	0.683372	0.746091	0.777605	0.795009
0.082032	0.684426	0.747561	0.779161	0.797386
0.099528	0.685649	0.748997	0.780607	0.799642
0.117039	0.686997	0.750400	0.781953	0.801742
0.134531	0.688416	0.751773	0.783214	0.803667
0.151981	0.689862	0.753099	0.784400	0.805450
0.169381	0.691338	0.754380	0.785515	0.807090
0.186729	0.692846	0.755618	0.786565	0.808598
0.204022	0.694402	0.756818	0.787554	0.809982
0.221256	0.696046	0.757986	0.788486	0.811247
0.238427	0.697854	0.759133	0.789362	0.812399
0.255529	0.699925	0.760259	0.790180	0.813438
0.272556	0.702364	0.761372	0.790945	0.814371
0.289495	0.705204	0.762466	0.791651	0.815196
0.306329	0.708367	0.763527	0.792283	0.815900
0.323028	0.711671	0.764536	0.792831	0.816473
0.339539	0.714897	0.765462	0.793266	0.816886
0.355772	0.717850	0.766253	0.793540	0.817092
0.371580	0.720379	0.766837	0.793580	0.817011
0.386730	0.722362	0.767098	0.793264	0.816512
0.400908	0.723687	0.766856	0.792404	0.815388
0.413741	0.724163	0.765851	0.790736	0.813351
0.424901	0.723528	0.763702	0.787929	0.810037
0.434221	0.721726	0.760292	0.783818	0.805293
0.441724	0.718791	0.755727	0.778520	0.799252
0.447591	0.715100	0.750572	0.772651	0.792578
0.452088	0.711513	0.745525	0.766924	0.786105
0.455487	0.708356	0.741149	0.761970	0.780502
0.458035	0.705798	0.737605	0.757955	0.776017
0.459936	0.703880	0.734910	0.754882	0.772539
0.461351	0.702389	0.732859	0.752560	0.770007
0.462400	0.701371	0.731393	0.750860	0.768053
0.463180	0.700507	0.730241	0.749572	0.766724
0.463757	0.700012	0.729485	0.748665	0.765619
0.464185	0.699465	0.728801	0.747930	0.764950
0.464502	0.699264	0.728440	0.747460	0.764281
0.464737	0.698880	0.727991	0.747005	0.763964
0.465883	0.696625	0.725152	0.744040	0.760689
0.468697	0.692664	0.719866	0.738240	0.753871
0.472963	0.686949	0.712053	0.729629	0.743935
0.478481	0.681016	0.703998	0.720789	0.733820
0.485048	0.676092	0.697626	0.713964	0.726173
0.492442	0.672743	0.693840	0.710320	0.722321

Table 5 Continued

Pitch	PR = 1.18	PR = 1.22	PR = 1.24	PR = 1.25
0.500407	0.670922	0.692752	0.709944	0.722516
0.508589	0.669046	0.691974	0.709884	0.722922
0.516688	0.667502	0.691692	0.710402	0.723981
0.524329	0.666238	0.691776	0.711275	0.725425
0.531099	0.665141	0.691868	0.712049	0.726670
0.536558	0.664211	0.691836	0.712544	0.727489
0.540257	0.663567	0.691719	0.712759	0.727827
0.541757	0.663144	0.691483	0.712655	0.727686
0.542018	0.663004	0.691353	0.712530	0.727425
0.542370	0.663118	0.691534	0.712758	0.727768
0.542844	0.663009	0.691490	0.712745	0.727721
0.543484	0.663080	0.691664	0.712983	0.728041
0.544347	0.663016	0.691728	0.713117	0.728201
0.545510	0.663051	0.691936	0.713427	0.728599
0.547073	0.663008	0.692130	0.713747	0.728996
0.549174	0.663021	0.692426	0.714213	0.729571
0.551985	0.662994	0.692762	0.714761	0.730232
0.555727	0.662976	0.693162	0.715431	0.731012
0.560662	0.662889	0.693572	0.716160	0.731820
0.567077	0.662800	0.693916	0.716876	0.732556
0.575232	0.662481	0.694148	0.717486	0.733119
0.585281	0.662207	0.694320	0.718002	0.733565
0.597194	0.661493	0.694792	0.718565	0.734092
0.610732	0.661191	0.695899	0.719368	0.734881
0.625415	0.661893	0.697406	0.720558	0.735905
0.640944	0.662571	0.698848	0.722262	0.737233
0.657007	0.663272	0.699700	0.724367	0.738811
0.673375	0.664034	0.701029	0.726508	0.740576
0.689905	0.664827	0.703192	0.728527	0.742467
0.706511	0.665663	0.705267	0.730407	0.744440
0.723142	0.666556	0.707289	0.732552	0.746471
0.739751	0.667506	0.709341	0.734977	0.748556
0.756337	0.668514	0.711422	0.737428	0.750735
0.772909	0.669577	0.713549	0.739911	0.753036
0.789464	0.670684	0.715733	0.742452	0.755443
0.806002	0.671829	0.717971	0.745046	0.757940
0.822524	0.673000	0.720260	0.747689	0.760519
0.839033	0.674179	0.722588	0.750372	0.763170
0.855531	0.675338	0.724944	0.753082	0.765886
0.872020	0.676454	0.727312	0.755804	0.768658
0.888481	0.677413	0.729593	0.758447	0.771449
0.904915	0.678243	0.731806	0.761024	0.774255
0.921301	0.678958	0.733942	0.763526	0.777061
0.937591	0.679573	0.735979	0.765924	0.779840
0.953691	0.680101	0.737890	0.768184	0.782555
0.969441	0.680577	0.739680	0.770300	0.785184
0.984586	0.681075	0.741282	0.772194	0.787599
1.000000	0.681763	0.742892	0.774036	0.789999

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C: DATA ACQUISITION SYSTEM

Figure 68 shows the connection of the Kulite transducer to the data acquisition system. The full bridge Kulite Pressure Transducer was connected to the Hewlett-Packard E1529A Remote Strain Conditioning Unit, [Ref. 7], via a RJ-45 cable. The HP E1422A, [Ref. 7], was connected, by RJ-45, to the data interface port of the HP E1529A. The HP E1529A was connected, from the 37-pin connection, to the HP E1433A high-speed digitizer, [Ref. 9]. The tachometer signal was also connected to the HP E1433A digitizer. The HP E1422A, E1433A, and Agilent N2216A were addressed through the HP E8404A VXI Mainframe, [Ref. 9]. The Kulite excitation voltage, 5 Volts, from an adjustable power supply, was input to the bridge excitation port of the HP E1529A. The VXI Mainframe was interfaced to a PC, with a 'firewire' interface. An Agilent program, DAC Express, was used to acquire data.

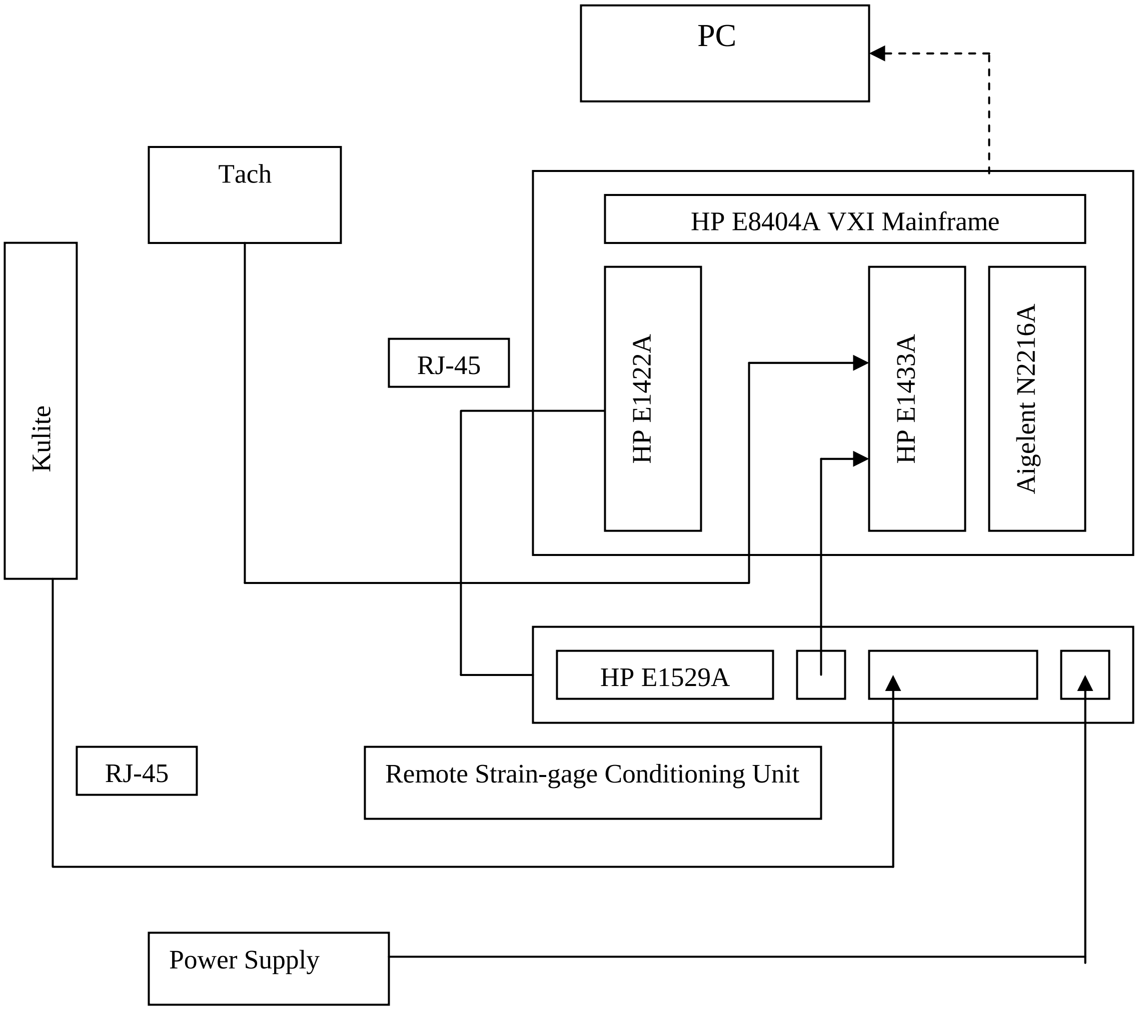


Figure 68. Data Acquisition System

Figure 69 shows the bridge excitation input of the HP E1529A connections to the dual-channel power supply. The Kulite required an excitation voltage of 5V. V1 and V2 of the power supply were set to 2.5V. A wire was connected from the negative of V1 and the positvie of V2 to ground. The positive from the bridge excitation, channel 16-23, was connected to the positive of V1; the negative of the bridge excitation, channel 16-23, was connected to the negative of V2.

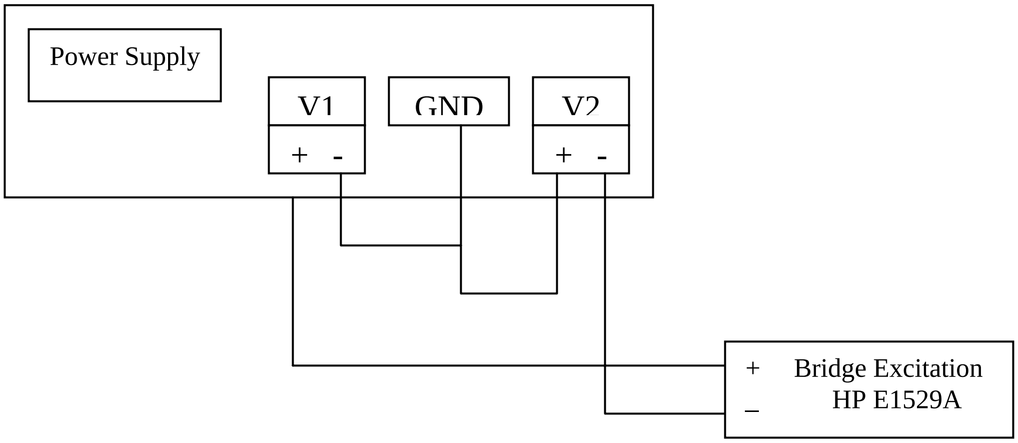


Figure 69. Bridge Excitation Power Supply Connection

Figure 70 shows the Kulite wires and their correct pin assignments. The Kulite Pressure Transducer had four wires, a green, white, black, and red wire. The green wire was connected to pin 3, the white to pin 6. The black wire was connected to pins 2 and 6, and the red to pins 1 and 8. Pins 4 and 5 were not connected to any wires. The RJ-45 connected the Kulite to the HP E1529A.

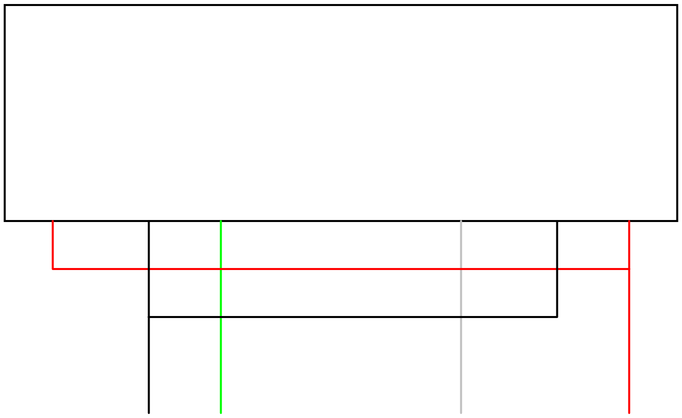
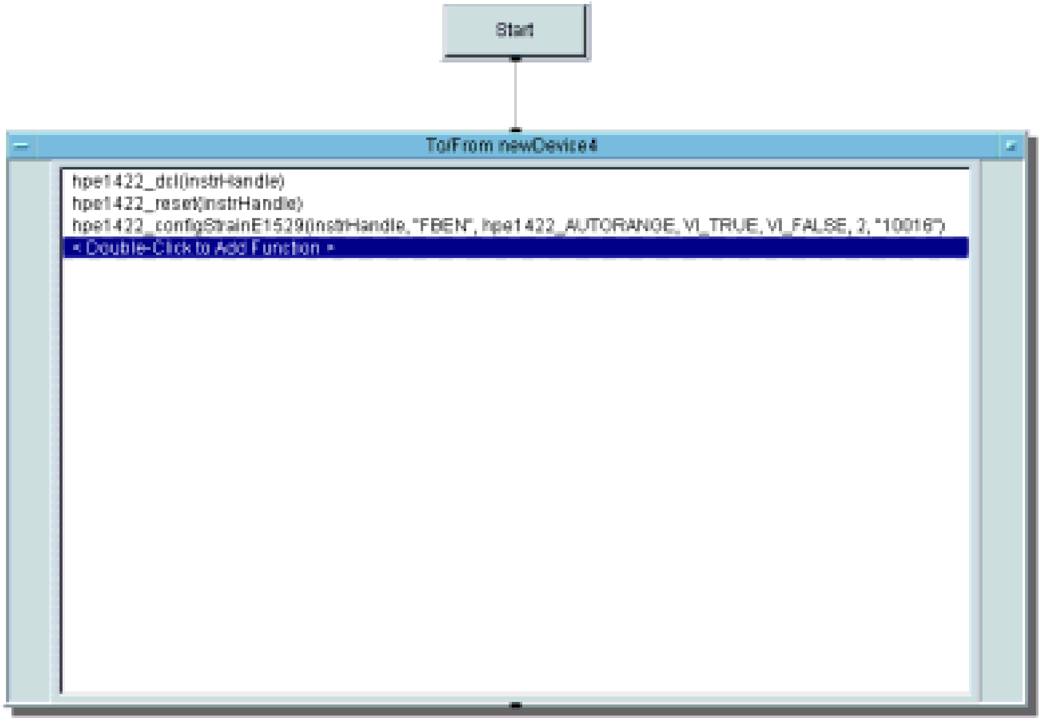


Figure 70. Kulite Connection to RJ-45 Plug

Figure 71 shows the HP Vee Pro GUI, [Ref. 8], which was used to set the HP E1529A, [Ref. 7], to a full bridge configuration. Figure 72 shows the full bridge configuration of the HP E1529A. Table 7 gives the factory specifications of the HP E1529A.



```
hpe1422_dcl(instrHandle)
hpe1422_reset(instrHandle)
hpe1422_configStrainE1529(instrHandle, "FBEN", hpe1422_AUTORANGE, VI_TRUE, VI_FALSE, 2, "10016")
« Double-Click to Add Function »
```

Figure 71. HP Vee Pro GUI

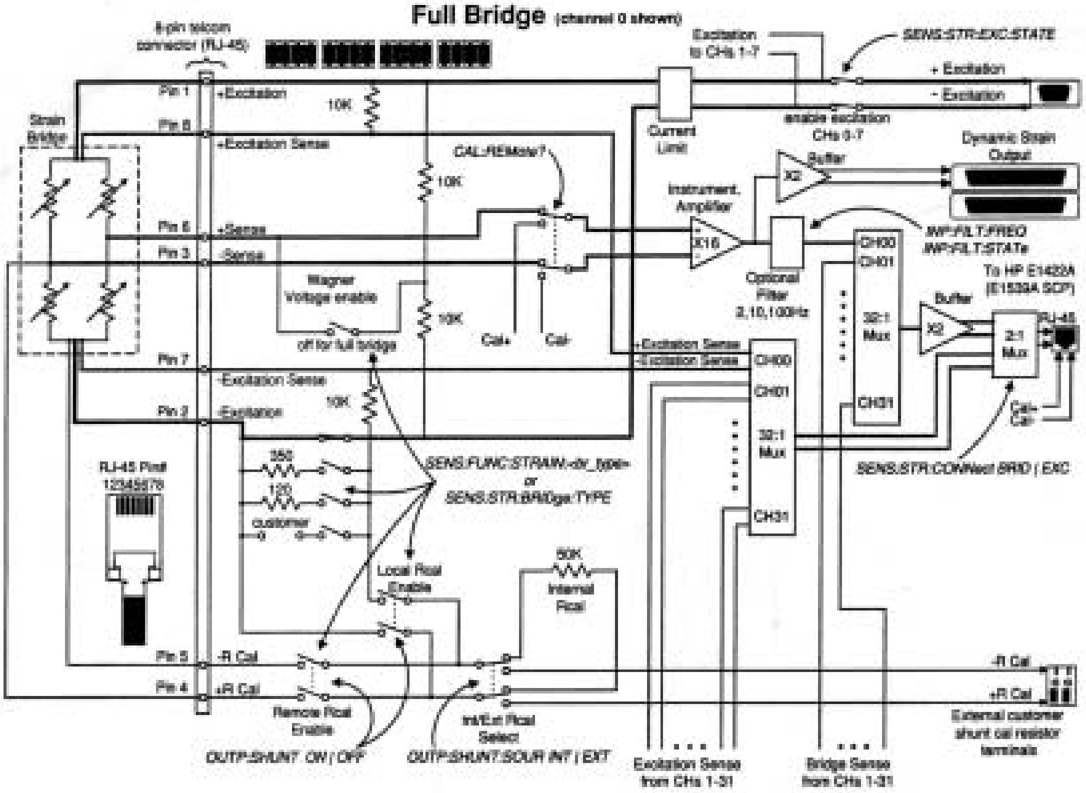


Figure 72. Full Bridge Configuration of the HP E1529A

Table 7 HP E1529A Specifications

General		
Outputs	32 individually buffered dynamic outputs	
Bridge Completion	120, 350 Ohm	
Bridge Configuration	Full, half, quarter	
Remote Operation	1000 ft	
Bridge Excitation	User-supplied	
Linearization	Mx+b	
Calibration	Internal self-calibration	
Measurement Rate	196 kSa/s dynamic	
Strain Measurement		
Voltage Offset	<2 microvolt	
Gain Error	<0.015%	
RMS Noise	<1 microvolt rms	
CMRR	>100dB	
Drift		
Offset Drift	<1 microvolt/degC	
Gain Drift	<30 ppm/degC	
Dynamic Outputs	32V per V	
Gain	<250 microvolt	
Offset	>20 kHz	
Full Bridge Bending Errors	Strain	Error
	με	με
	-50000	28
	-40000	22
	-30000	17
	-20000	11
	-10000	6
	0	0.3
	10000	6
	20000	11
	30000	17
	40000	22
	50000	28

A. DAC EXPRESS

Hewlett-Packard DAC Express, [Ref. 11], was used to display the digitized signal in real time. Figure 73 shows the DAC Express GUI as configured for the unsteady pressure measurements. The DAC Express GUI was configured to plot both the digitized tach voltage and the digitized unsteady voltage. A virtual dial gauge was placed on the GUI to display the speed of the compressor. Figure 74 shows the channel inputs for DAC Express. Channel 1 and Channel 13 were selected for recording, Channel 1 was the unsteady voltage and Channel 13 was the tachometer voltage. The range of the A/D converter was set to 2V and 10V for Channel 1 and 13, respectively. Figure 75 shows the timebase selected in DAC Express for the unsteady pressure measurements. The timebase was set to trigger on a positive slope of the tachometer voltage at 2.5 Volts. For dynamic measurements the sampling rate was set to 196608 samples/sec with a sampling period of 5.09E-06 and a blocksize of 256 samples.

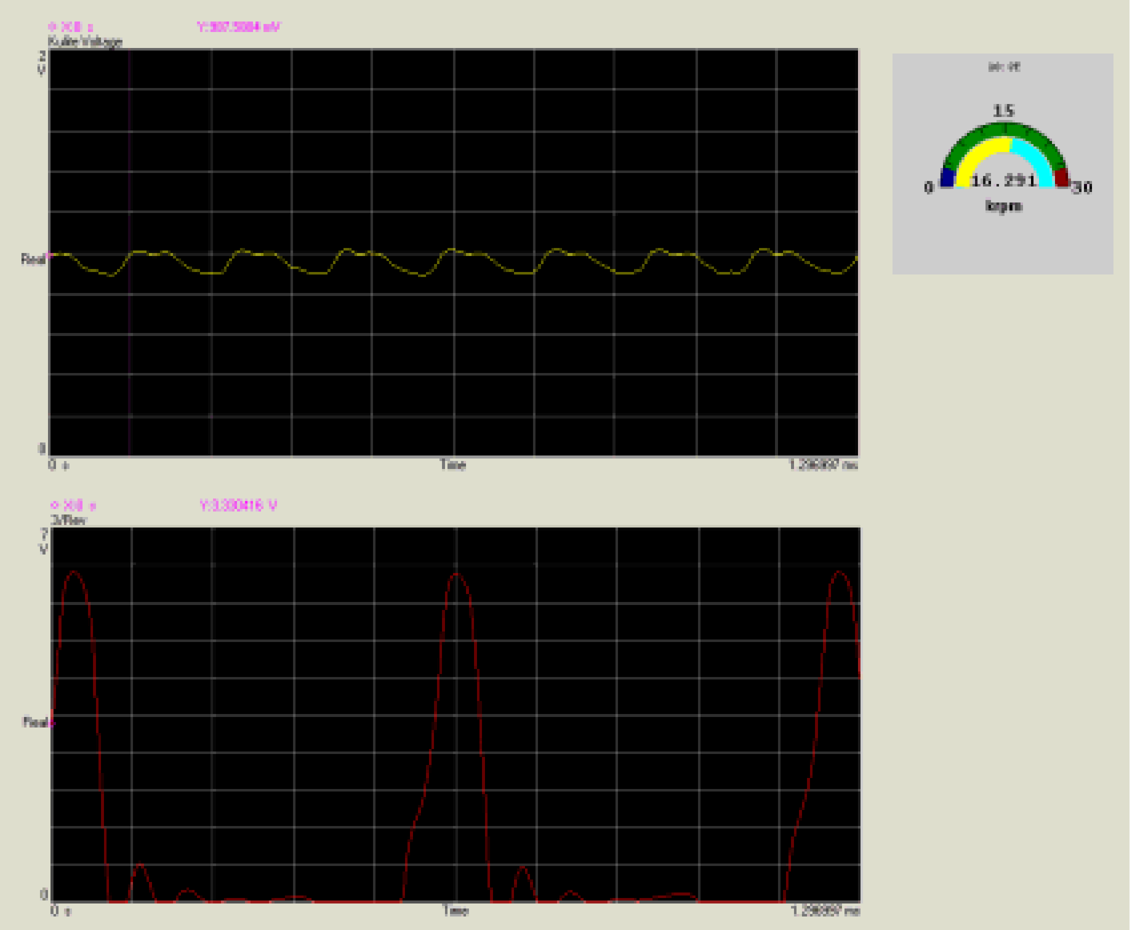
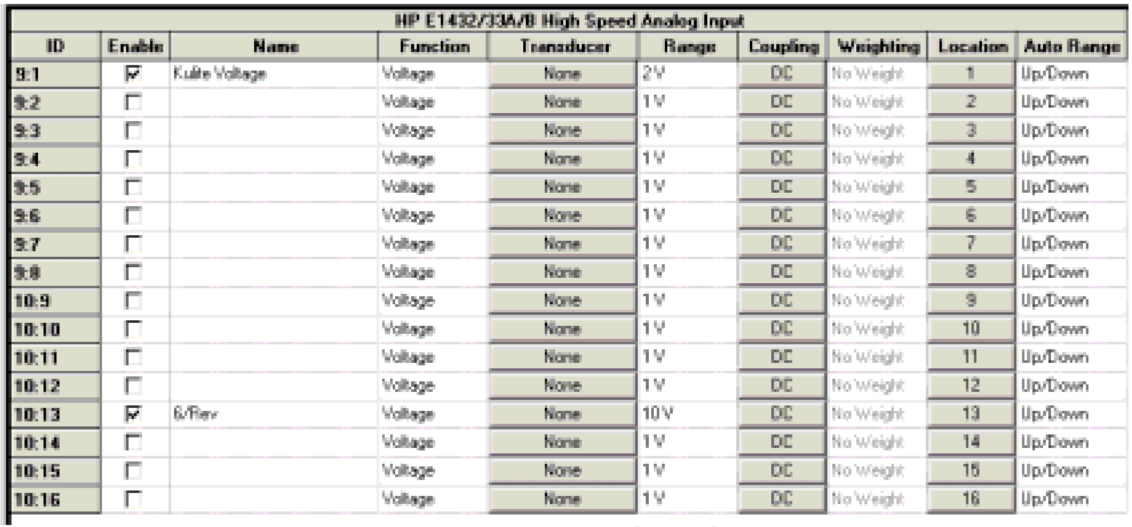
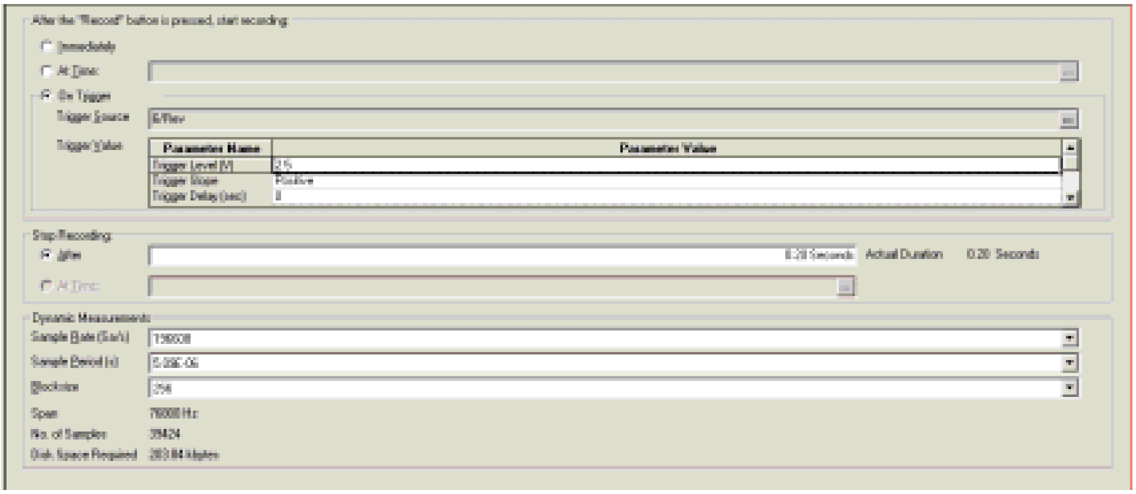


Figure 73. DAC Express GUI



HP E1432/33A/8 High Speed Analog Input									
ID	Enable	Name	Function	Transducer	Range	Coupling	Weighting	Location	Auto Range
9:1	☑	Kulite Voltage	Voltage	None	2 V	DC	No Weight	1	Up/Down
9:2	□		Voltage	None	1 V	DC	No Weight	2	Up/Down
9:3	□		Voltage	None	1 V	DC	No Weight	3	Up/Down
9:4	□		Voltage	None	1 V	DC	No Weight	4	Up/Down
9:5	□		Voltage	None	1 V	DC	No Weight	5	Up/Down
9:6	□		Voltage	None	1 V	DC	No Weight	6	Up/Down
9:7	□		Voltage	None	1 V	DC	No Weight	7	Up/Down
9:8	□		Voltage	None	1 V	DC	No Weight	8	Up/Down
10:9	□		Voltage	None	1 V	DC	No Weight	9	Up/Down
10:10	□		Voltage	None	1 V	DC	No Weight	10	Up/Down
10:11	□		Voltage	None	1 V	DC	No Weight	11	Up/Down
10:12	□		Voltage	None	1 V	DC	No Weight	12	Up/Down
10:13	☑	6/Rev	Voltage	None	10 V	DC	No Weight	13	Up/Down
10:14	□		Voltage	None	1 V	DC	No Weight	14	Up/Down
10:15	□		Voltage	None	1 V	DC	No Weight	15	Up/Down
10:16	□		Voltage	None	1 V	DC	No Weight	16	Up/Down

Figure 74. DAC Express Channel Input



After the "Record" button is pressed, start recording:

○ Immediately

○ At Time:

● On Trigger

Trigger Source: 6/Rev

Trigger Value:

Parameter Name	Parameter Value
Trigger Level (V)	2.5
Trigger Slope	Positive
Trigger Delay (sec)	0

Stop Recording

● After: 0.20 Seconds Actual Duration 0.20 Seconds

○ At Time:

Dynamic Measurements

Sample Rate (Sa/s): 196608

Sample Period (s): 5.09E-06

Blocksize: 256

Span: 76800 Hz

No. of Samples: 39424

Disk Space Required: 203.84 kbytes

Figure 75. DAC Express Timebase

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D: CALIBRATION OF THE KULITE AND REDUCTION OF DATA

Initially, the Kulite was set up and recording procedures were verified using an unsteady flow generator, as shown in Figure 76. The generator consisted of a wheel, which had 60 holes around its periphery that allowed pressurized air from a jet to intermittently impact the face of the Kulite. As the wheel spun it chop the pressurized air jet, creating an unsteady pressure. The wheel had an additional three holes; at a smaller radius. An integrated fiber optic laser probe detected the holes, giving a three per rev signal. With the wheel spinning at a speed of 10,000 rpm, the Kulite saw the equivalent of a rotor with 20 blades spinning at 30,000 rpm, the approximate characteristics of the compressor.

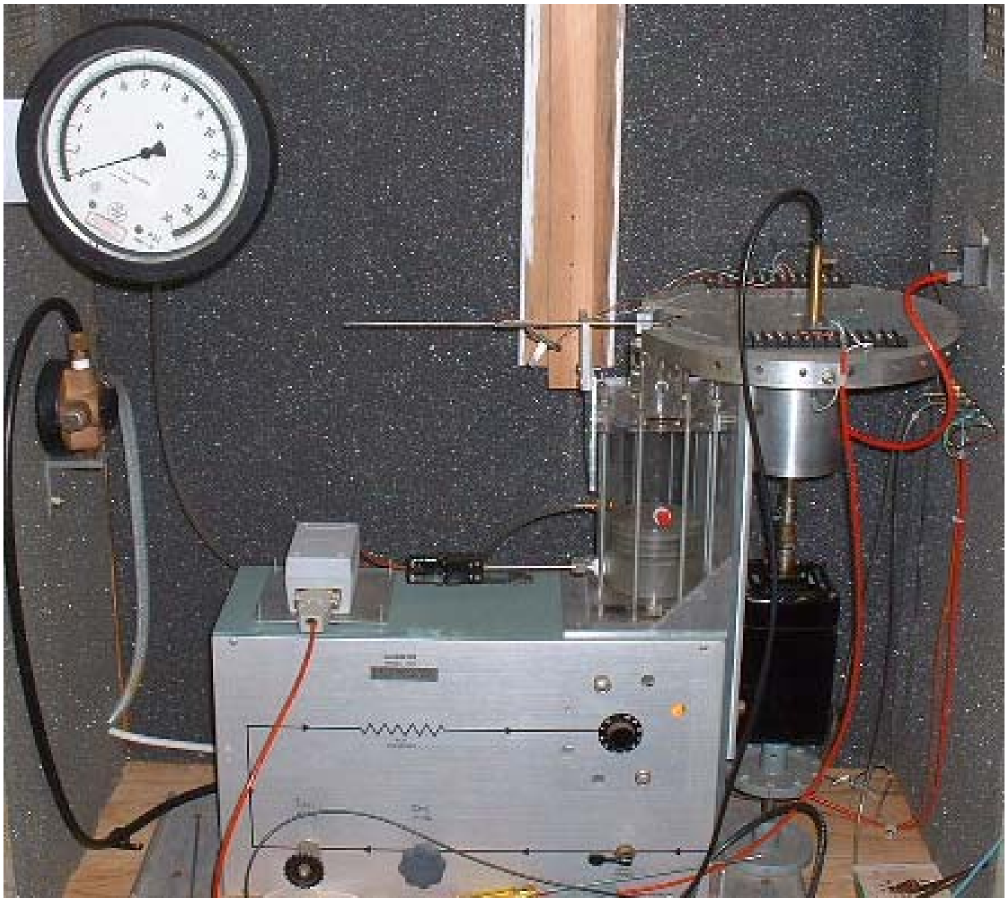


Figure 76. Wheel For Initial Kulite Testing

The data collected by DAC Express, from the wheel, were used to validate the calibration and data analysis programs. The characteristics of the data were not expected to duplicate the unsteady pressure in the compressor. The blade to blade differences, as well as the standard deviation of multiple revolutions were large, due to the grossly unsteady process generated by chopping a free-jet. MATLAB 5.3, [Ref. 12], was used due to incompatibility with newer versions of MATLAB; specifically toolboxes needed for calibration and data reduction.

A. CALIBRATION

The calibration of the Kulite used three MATLAB scripts. The voltage output of the Kulite was linearly proportional to the pressure across the sensor. For data reduction, voltage was the input and pressure was the output, the calibration was written as

$$\overline{P} = a + b * \overline{V} \tag{1}$$

P bar was the time-averaged differential pressure and V bar was the time-averaged voltage; a was the intercept and b was the slope of the calibration curve. P bar was given by

$$\overline{P} = \overline{P_S} - P_{REF} \tag{2}$$

Where P_S bar was the time average of the unsteady pressure on the transducer face and P_{REF} was the reference pressure applied to the back of the Kulite. Substituting equation (2) into equation (1),

$$\overline{P_S} - P_{REF} = a + b * \overline{V} \tag{3}$$

was the equation used for the calibration.

The calibration procedure involved recording voltage data using DAC Express, when four different reference pressures (P_{REF}) where applied; namely, 0, 5, 10, and 15 inches of mercury gauge respectively. The time-average of the sensed pressure (P_S bar) was known, and was input with the reduction. A MATLAB script, readsdftd, was installed with the DAC Express toolbox, [Ref. 11], which took the exported .sdf file and read it to a vector in MATLAB. Averaging the unsteady voltage gave V bar; P_S bar and P_{REF} were inputs. MATLAB's polyfit command, [Ref. 12], was used to fit the data to a polynomial, in this case to a straight line. The intercept, a, and the slope, b, were saved and used as inputs for data reduction.

The unsteady pressure differential can be written as a function of unsteady voltage as

$$P = a + b*V \tag{4}$$

The unsteady differential pressure is given by the unsteady sensor pressure minus the steady reference pressure, or

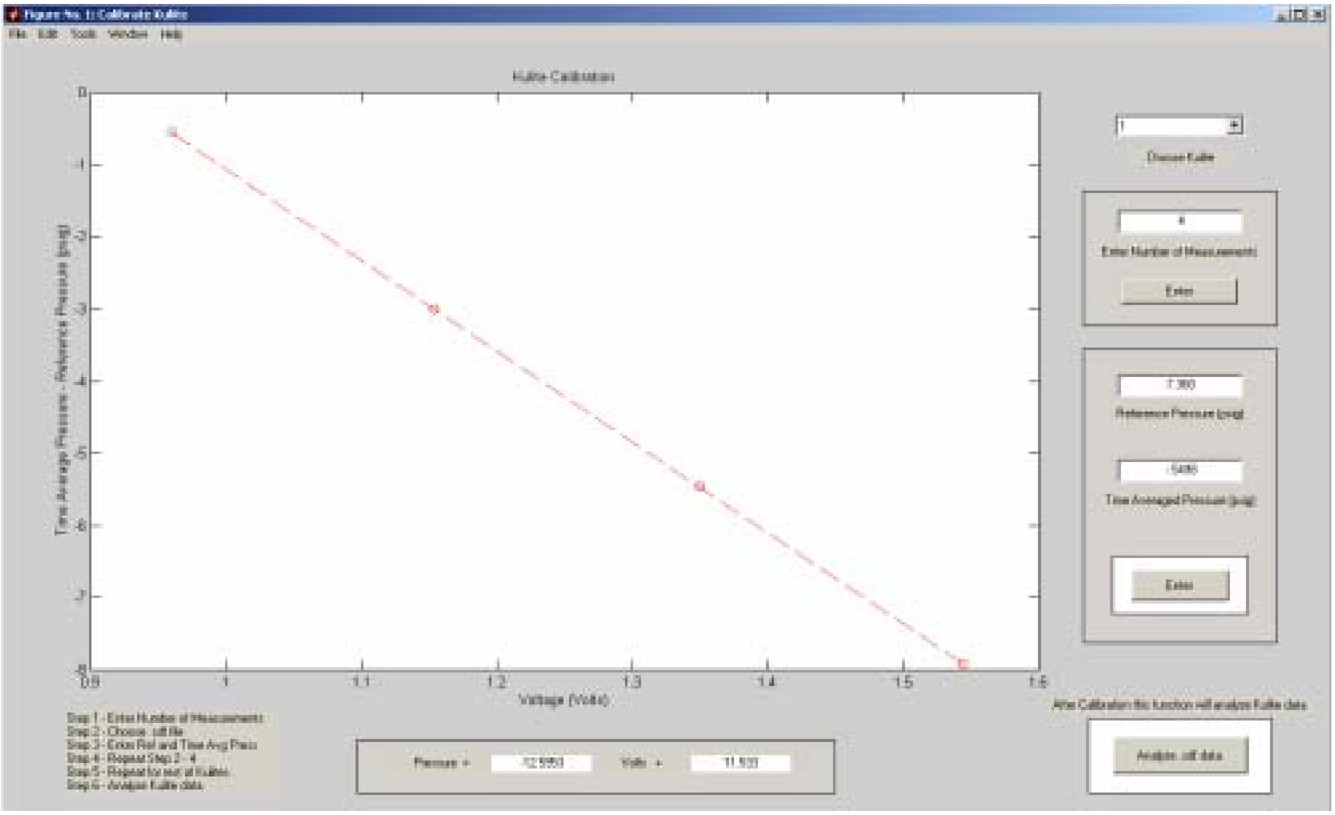
$$P = P_S - P_{REF} \tag{5}$$

The reference (gauge) pressure, for the data set reduced to unsteady pressure, was zero. Taking this into account, and substituting equation (5) into equation (4) the equation used in data reduction became

$$P_S = a + b*V \tag{6}$$

where P_S is the unsteady gauge pressure.

Figure 77 shows the MATLAB GUI, [Ref. 12], used in calibration of the Kulite.



1

Enter Number of Measurements: 4

Reference Pressure (psig): 7.360

Time Averaged Pressure (psig): -.5486

After Calibration this function will analyze Kulite data

Step 1 - Enter Number of Measurements
Step 2 - Choose .sdf file
Step 3 - Enter Ref and Time Avg Press
Step 4 - Repeat Step 2 - 4
Step 5 - Repeat for rest of Kulites
Step 6 - Analyze Kulite data

Pressure = -12.5950 Volts = 11.533

Figure 77. Calibration GUI

B. DATA REDUCTION

Analysis of the data read by the Kulite used five MATLAB scripts. The exported .sdf data file, once read by readsdftd [Ref. 11], had a scalar, FS, and a matrix, Y. The Y matrix contained the list of raw data, digitized unsteady voltages, and there were two columns. The first column was the Kulite output voltage, and the second column was the tachometer signal voltage. The scalar, FS, was the sampling rate used to take the data. The sampling rate was used to set up a time vector, t(i), to assign to the tachometer voltage, v(i). Note that, at 60% design speed there were 40,000 points between 0 and .2 seconds. The time vector and the tachometer voltage vector were used to create the vectors tp1 and vp1. The vectors, tp1 and vp1, were related by tp1(i+1) = t(i), and vp1(i+1) = v(i). MATLAB’s find function, [Ref. 12], was used to find the location, Loc, of v(Loc) that was less than the trigger and vp1(Loc) that was greater than the trigger, Loc = find(v<Trig and vp1>Trig). The location was the start of each revolution and was used to find the start time of each revolution, given by

$$time(Loc) = t(Loc) + \frac{trig - v(Loc)}{vp1(Loc) - v(Loc)} * (tp1(Loc) - t(Loc)) \quad (7)$$

The time of each revolution was used to find the velocity of each revolution, or

$$vel(Loc) = \frac{\pi * D}{t(Loc+1) - t(Loc)} \quad (8)$$

The distance over the rotor was written as

$$dis(i) = vel(Loc) * (t(Loc+1) - t(Loc)) \quad (9)$$

which assigned the unsteady voltage measurements to a specific spot on the rotor. The distance from equation (9) was non-dimensionalized by

$$dis_{(Non-Dim)} = \frac{22 * dis}{\pi * D} \quad (10)$$

The non-dimensionalized distance was the x-axis on the graph of unsteady pressure over the rotor. The x-axis ranged from zero to twenty-two, for each blade on the rotor. The intercept and slope were saved from the calibration of the Kulite and used to convert the unsteady voltage to pressure, written as

$$P_S = a + b * V \quad (11)$$

The unsteady pressure found in equation (11) was converted to a non-dimensional pressure ratio, given by

$$P_{S(Non-Dim)} = \frac{P_S + P_{amb}}{P_{tot} * \gamma} \quad (12)$$

where P_{tot} was the upstream total pressure, P_{amb} was the ambient pressure, and γ was the ratio of specific heats. The pressure ratio was the y-axis on the graph of unsteady pressure over the rotor, and was the parameter output by the CFD computations.

Figure 78 shows the MATLAB GUI, [Ref. 12], used to implement the data analysis.

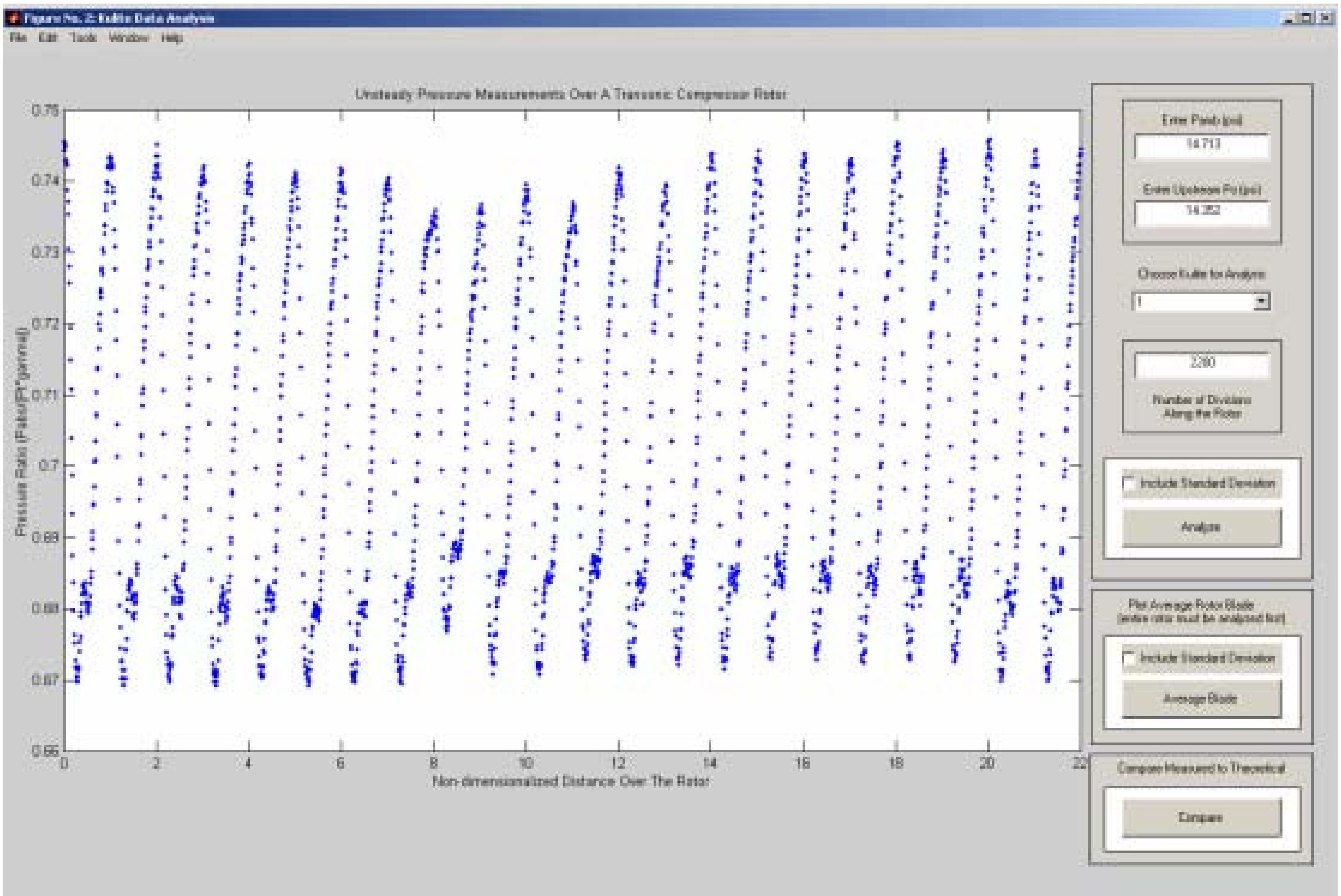


Figure No. 2: Kulite Data Analysis
File
Edit
Tools
Window
Help
Unsteady Pressure Measurements Over A Transonic Compressor Rotor
Pressure Ratio (Pabs/(Pt*gamma))
0.75
0.74
0.73
0.72
0.71
0.7
0.69
0.68
0.67
0.66
0
2
4
6
8
10
12
14
16
18
20
22
Non-dimensionalized Distance Over The Rotor
Include Standard Deviation
Analyze
Plot Average Rotor Blade
(entire rotor must be analyzed first)
Include Standard Deviation
Average Blade
Compare Measured to Theoretical
Compare

Enter Pamb (psi): 14.713

Enter Upstream Po (psi): 14.352

Choose Kulite for Analysis: 1

2280 Number of Divisions Along the Rotor

□ Include Standard Deviation

□ Include Standard Deviation

Figure 78. Data Analysis GUI

APPENDIX E: MATLAB CODE

A. CALIBRATION

1. Calibrate.m

```
%
%  This function calibrates the kulite by averaging the voltage reading
%  of a .sdf file for a given reference pressure
%
%  Matlab files needed for running this m-file
%    calibrate_1.m
%    calibrate_2.m
%
function fig = calibrate()
% This is the machine-generated representation of a Handle Graphics
% object and its children.  Note that handle values may change when
% these objects are re-created. This may cause problems with any
% callbacks written to depend on the value of the handle at the time
% the object was saved.  This problem is solved by saving the output as
% a FIG-file.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.
%
% NOTE: certain newer features in MATLAB may not have been saved in
% this M-file due to limitations of this format, which has been
% superseded by FIG-files.  Figures which have been annotated using the
% plot editor tools are incompatible with the M-file/MAT-file format,
% and should be saved as FIG-files.

load calibrate

h0 = figure('Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'FileName','C:\Documents and 
Settings\Administrator\Desktop\Matt\calibrate.m', ...
   'Name','Calibrate Kulite', ...
   'PaperPosition',[18 180 576 432], ...
   'PaperUnits','points', ...
   'Position',[4 242 1274 744], ...
   'Tag','Fig1', ...
   'ToolBar','none');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[781.875 15.75 132.75 54.75], ...
   'Style','frame', ...
   'Tag','Anal_Frame');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804 
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[778.125 354 140.25 98.25], ...
   'Style','frame', ...
   'Tag','Num_Frame');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[778.125 124.5 140.25 213.75], ...
   'Style','frame', ...
   'Tag','Press_Frame');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'ListboxTop',0, ...
   'Position',[254.25 15.75 345.75 39], ...
   'Style','frame', ...
   'Tag','Eqn_Frame');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[799.125 145.5 98.25 42.75], ...
   'Style','frame', ...
   'Tag','Cal_Frame');
h1 = axes('Parent',h0, ...
   'CameraUpVector',[0 1 0], ...
   'CameraUpVectorMode','manual', ...
   'Color',[1 1 1], ...
   'ColorOrder',mat1, ...
   'Position',[0.06514913657770802 0.1881720430107527
0.7166405023547882 0.7500000000000001], ...
   'Tag','Cal_Graph', ...
   'XColor',[0 0 0], ...
   'YColor',[0 0 0], ...
   'ZColor',[0 0 0]);
h2 = text('Parent',h1, ...
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','center', ...
   'Position',[0.4989035087719299 -0.04308797127468589
9.160254037844386], ...
   'Tag','Axes1Text4', ...
   'VerticalAlignment','cap');
set(get(h2,'Parent'),'XLabel',h2);
h2 = text('Parent',h1, ...
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','center', ...
   'Position',[-0.03179824561403508 0.4973070017953322
9.160254037844386], ...
   'Rotation',90, ...
   'Tag','Axes1Text3', ...
   'VerticalAlignment','baseline');
set(get(h2,'Parent'),'YLabel',h2);
h2 = text('Parent',h1, ...
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','right', ...
   'Position',[-0.09210526315789473 1.080789946140036
9.160254037844386], ...
   'Tag','Axes1Text2', ...
   'Visible','off');
set(get(h2,'Parent'),'ZLabel',h2);
h2 = text('Parent',h1, ...
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','center', ...
   'Position',[0.4989035087719299 1.012567324955117 9.160254037844386],
...
   'Tag','Axes1Text1', ...
   'VerticalAlignment','bottom');
set(get(h2,'Parent'),'Title',h2);
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[802.875 302.25 90.75 17.25], ...
   'String','0', ...
   'Style','edit', ...
   'Tag','PRef_Edit');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[802.875 240.75 90.75 17.25], ...
   'String','0', ...
   'Style','edit', ...
   'Tag','Time_Avg_Edit');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[788.25 222 120 11.25], ...
   'String','Time Averaged Pressure (psig)', ...
   'Style','text', ...
   'Tag','Time_Avg');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[794.25 285 108 11.25], ...
   'String','Reference Pressure (psig)', ...
   'Style','text', ...
   'Tag','PRef');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'Callback','calibrate_2', ...
   'ListboxTop',0, ...
   'Position',[813.375 155.25 69.75 22.5], ...
   'String','Enter', ...
   'Tag','Press_Enter');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[279.75 31.5 72 12], ...
   'String','Pressure  =', ...
   'Style','text', ...
   'Tag','Pressure');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[351.75 31.5 72 12], ...
   'Style','text', ...
   'Tag','slope');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[423.75 31.5 72 12], ...
   'String','Volts   +', ...
   'Style','text', ...
   'Tag','Volts');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[495.75 31.5 72 12], ...
   'Style','text', ...
   'Tag','intercept');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[802.875 420.75 90.75 17.25], ...
   'Style','edit', ...
   'Tag','Num_Edit');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[790.5 403.5 115.5 9.75], ...
   'String','Enter Number of Measurements', ...
   'Style','text', ...
   'Tag','Num');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'Callback','calibrate_1', ...
   'ListboxTop',0, ...
   'Position',[806.25 370.5 84 18], ...
   'String','Enter', ...
   'Tag','Num_Enter');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[43.5 15 154.5 60.75], ...
   'String',mat2, ...
   'Style','text', ...
   'Tag','Instructions');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'Callback','Kulite_Analysis', ...
   'ListboxTop',0, ...
   'Position',[799.875 30 96.75 27], ...
   'String','Analyze .sdf data', ...
   'Tag','Kulite_Analysis');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[753 74.25 190.5 11.25], ...
   'String','After Calibration this function will analyze Kulite data',
...
   'Style','text', ...
   'Tag','K_Analysis');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[801.375 488.25 93.75 18.75], ...
   'String',['1';'2';'3';'4';'5';'6'], ...
   'Style','popupmenu', ...
   'Tag','Cal_Kulite', ...
   'Value',1);
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[808.125 470.25 80.25 11.25], ...
   'String','Choose Kulite', ...
   'Style','text', ...
   'Tag','Choose');
if nargout > 0, fig = h0; end
```

2. Calibrate_1.m

```
%
%  This m file is used as part of Calibrate.m
%
cla
%
```

```
Cal_Num = 6;      %Number of Kulites to calibrate
%
Pcal = zeros(1,Cal_Num);
Vcal = zeros(1,Cal_Num);
slope = zeros(Cal_Num,1);
inter = zeros(Cal_Num,1);
%
i = 1;
%
meas = findobj('Tag','Num_Edit');
meas = get(meas,'String');
meas = str2num(meas);
len = meas;
meas = meas + 1;
%
if meas <= 2;
   plot(0,0)
   text(-.25,0,'Must have at least two measurements')
   error('Must have at least two measurements')
elseif size(meas) == 0;
   plot(0,0)
   text(-.25,0,'Must have at least two measurements')
   error('Must have at least two measurements')
else
   cla
end
%
[newfile,newpath] = uigetfile('*.sdf','Select File');
SDF.filename = [newpath newfile];
if newfile ~= 0;
   [Y, FS] = readsdftd([newpath newfile]);
   cla
else
   plot(0,0)
   text(-.15,0,'Please try again')
   error('Please try again')
end
```

3. Calibrate_2.m

```
%
%  This m file is used as part of Calibrate.m
%
if meas == 0;
   cla
   plot(0,0)
   text(-.25,0,'First enter number of measurements')
   error('First enter number of measurements')
elseif size(meas) == 0;
   cla
   plot(0,0)
   text(-.25,0,'First enter number of measurements')
   error('First enter number of measurements')
else
   cla
end
```

```
%
cal_num = findobj('Tag','Cal_Kulite');
cal_num = get(cal_num,'Value');
%
ref = findobj('Tag','PRef_Edit');
ref = get(ref,'String');
ref = str2num(ref);
%
avg = findobj('Tag','Time_Avg_Edit');
avg = get(avg,'String');
avg = str2num(avg);
%
Vcal(i,cal_num) = mean(Y(:,cal_num));
%
if ref <= 0;
   Pcal(i,cal_num) = avg + ref;
else
   Pcal(i,cal_num) = avg - ref;
end
%
i = i + 1;
meas = meas - 1;
%
%     Plot
%
if meas == 1;
   plot(Vcal(1:len,cal_num),Pcal(1:len,cal_num),'ro')
   title('Kulite Calibration');
   xlabel('Voltage (Volts)');
   ylabel('Time Averaged Pressure - Reference Pressure (psig)');
   %
   [coeff S] = polyfit(Vcal(1:len,cal_num),Pcal(1:len,cal_num),1);
   slope(cal_num,1) = coeff(1,1);
   inter(cal_num,1) = coeff(1,2);
   %
   int = findobj('Tag','intercept');
   set(int,'String',inter(cal_num,1));
   %
   slop = findobj('Tag','slope');
   set(slop,'String',slope(cal_num,1));
   %
   [J,K] = size(Pcal(1:len,cal_num));
   Press = zeros(J,1);
   one = ones(J,1);
   Press = (slope(cal_num,1).*Vcal(1:len,cal_num)) +
inter(cal_num,1).*one;
   %
   hold on;
   plot(Vcal(1:len,cal_num),Press,'r--')
else
   [newfile,newpath] = uigetfile('*.sdf','Select File');
   SDF.filename = [newpath newfile];
   if newfile ~= 0;
      [Y, FS] = readsdftd([newpath newfile]);
      cla
   else
      i = i - 1;
      meas = meas + 1;
      calibrate_2
   end
end
save Slope.txt slope -ascii;
save Intercept.txt inter -ascii;
```

B. DATA REDUCTION

1. Kulite_Analysis.m

```
%  Kulite_Analysis
%
%  This program analyzes the data obtained in DACExpress for the
%  Kulite.  The data is converted into pressure given the calibration
%  equation, and then plotted versus distance along the rotor.
%
%  Matlab files needed for running this m-file
%    analyze.m
%    analyze_1.m
%    analyze_2.m
%    analyze_3.m
%
%  Set Variables
%
clear all
%
Num = 1;       %Number of Kulites
%
D = 11.02;     %Inner Case Wall Diameter in Inches
per = 6;       %Number of per revolutions for the tach
Trig = 2.0;    %Trigger Level in Volts
%
%  Obtain SDF File
%
[newfile,newpath] = uigetfile('*.sdf','Select File');
SDF.filename = [newpath newfile];
if newfile ~= 0;
   [Y, FS] = readsdftd([newpath newfile]);
   cla
else
   Kulite_Analysis
end
%
%  Initialize Matrices
%
m = size(Y,1);
Time = [1/FS:1/FS:m/FS]';     %time of sample
Z = zeros(m,Num+2);
X = [Time,Y,Z];
%
%  Establish Start of Revolution
%
tach = 1+Num;                         %Tach Channel
v = X(1:(m-1),(tach+1));
vp1 = X(2:m,(tach+1));
```

```
t = X(1:(m-1),(1));
tp1 = X(2:m,(1));
%
Loc = find( v<Trig & vp1>Trig );      %location of time of start of rev
M = size(Loc,1);
count = M/per;
count = floor(count);
Rev = count - 1;               %Maximum number of revolutions
i = 0;
while count > 0;
   temp(i+1,1) = Loc((i*per)+1,1);
   count = count - 1;
   i = i +1;
end
Loc = temp;
time = zeros(Rev,1);
for i = 1:Rev+1;
%interpolate to find
    time(i) = X(Loc(i),1) + (Trig - v(Loc(i)))/(vp1(Loc(i)) -
v(Loc(i)))*(tp1(Loc(i)) - t(Loc(i)));     %exact time of start of rev
end
%
%  Run Analyze gui
%
analyze
```

2. Analyze.m

```
%
% This is the gui written for Kulite_Analysis
%
function fig = analyze()
% This is the machine-generated representation of a Handle Graphics
% object and its children.  Note that handle values may change when
% these objects are re-created. This may cause problems with any
% callbacks written to depend on the value of the handle at the time
% the object was saved.  This problem is solved by saving the output as
% a FIG-file.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.
%
% NOTE: certain newer features in MATLAB may not have been saved in
% this M-file due to limitations of this format, which has been
% superseded by FIG-files.  Figures which have been annotated using the
% plot editor tools are incompatible with the M-file/MAT-file format,
% and should be saved as FIG-files.

load analyze

h0 = figure('Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'FileName','C:\Documents and
Settings\Administrator\Desktop\Matt\analyze.m', ...
   'Name','Kulite Data Analysis', ...
   'PaperPosition',[18 180 576 432], ...
   'PaperUnits','points', ...
   'Position',[2 166 1277 816], ...
   'Tag','Fig1', ...
   'ToolBar','none');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'ListboxTop',0, ...
   'Position',[780.75 23.25 161.25 80.25], ...
   'Style','frame', ...
   'Tag','Frame5');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'ListboxTop',0, ...
   'Position',[781.5 110.25 160.5 111.75], ...
   'Style','frame', ...
   'Tag','Frame4');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'ListboxTop',0, ...
   'Position',[781.5 228.75 159.75 357.75], ...
   'Style','frame', ...
   'Tag','Frame3');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'ListboxTop',0, ...
   'Position',[804.375 471 114 104.25], ...
   'Style','frame', ...
   'Tag','Frame1');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[790.875 243.75 141 73.5], ...
   'Style','frame', ...
   'Tag','Anal_Frame');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[811.125 419.25 100.5 18], ...
   'String',['1';'2';'3';'4';'5';'6'], ...
   'Style','popupmenu', ...
   'Tag','Kulite', ...
   'Value',1);
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[808.875 443.25 105 11.25], ...
   'String','Choose Kulite for Analysis', ...
   'Style','text', ...
   'Tag','Choose');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
```

```
   'Callback','analyze_1', ...
   'ListboxTop',0, ...
   'Position',[804 252.75 114.75 27.75], ...
   'String','Analyze', ...
   'Tag','Analyze');
h1 = axes('Parent',h0, ...
   'CameraUpVector',[0 1 0], ...
   'CameraUpVectorMode','manual', ...
   'Color',[1 1 1], ...
   'ColorOrder',mat1, ...
   'Position',[0.04463586530931871 0.1715686274509804
0.7642913077525451 0.7561274509803921], ...
   'Tag','Axes1', ...
   'XColor',[0 0 0], ...
   'YColor',[0 0 0], ...
   'ZColor',[0 0 0]);
h2 = text('Parent',h1, ...
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','center', ...
   'Position',[0.4994871794871795 -0.03896103896103886
9.160254037844386], ...
   'Tag','Axes1Text4', ...
   'VerticalAlignment','cap');
set(get(h2,'Parent'),'XLabel',h2);
h2 = text('Parent',h1, ...
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','center', ...
   'Position',[-0.02974358974358974 0.4983766233766235
9.160254037844386], ...
   'Rotation',90, ...
   'Tag','Axes1Text3', ...
   'VerticalAlignment','baseline');
set(get(h2,'Parent'),'YLabel',h2);
h2 = text('Parent',h1, ...
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','right', ...
   'Position',[-0.05948717948717949 1.094155844155844
9.160254037844386], ...
   'Tag','Axes1Text2', ...
   'Visible','off');
set(get(h2,'Parent'),'ZLabel',h2);
h2 = text('Parent',h1, ...
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','center', ...
   'Position',mat2, ...
   'Tag','Axes1Text1', ...
   'VerticalAlignment','bottom');
set(get(h2,'Parent'),'Title',h2);
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[804.375 334.5 114 67.5], ...
   'Style','frame', ...
   'Tag','Frame2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[812.625 373.5 97.5 19.5], ...
   'Style','edit', ...
   'Tag','divisions');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[824.625 340.5 73.5 23.25], ...
   'String','Number of Divisions Along the Rotor', ...
   'Style','text', ...
   'Tag','Divisions');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[803.625 287.25 115.5 21.75], ...
   'String','Include Standard Deviation', ...
   'Style','checkbox', ...
   'Tag','STD');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[812.625 483 97.5 19.5], ...
   'Style','edit', ...
   'Tag','Ptot_edit');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[812.625 531 97.5 19.5], ...
   'Style','edit', ...
   'Tag','Pamb_edit');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[808.875 554.25 105 11.25], ...
   'String','Enter Pamb (psi)', ...
   'Style','text', ...
   'Tag','Pamb');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[808.875 504 105 11.25], ...
   'String','Enter Upstream Po (psi)', ...
   'Style','text', ...
   'Tag','Ptotal');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[790.875 120.75 141 68.25], ...
   'Style','frame', ...
   'Tag','Aver_Frame');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[803.625 161.25 115.5 21.75], ...
   'String','Include Standard Deviation', ...
   'Style','checkbox', ...
   'Tag','std');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'Callback','analyze_2', ...
   'ListboxTop',0, ...
   'Position',[804 129 114.75 27.75], ...
   'String','Average Blade', ...
   'Tag','Average');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[797.25 192.75 128.25 23.25], ...
   'String','Plot Average Rotor Blade        (entire rotor must be
analyzed first)', ...
   'Style','text', ...
   'Tag','Plot');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[1 1 1], ...
   'ListboxTop',0, ...
   'Position',[790.875 33 141 46.5], ...
   'Style','frame', ...
   'Tag','Com_Frame');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'Callback','analyze_3', ...
   'ListboxTop',0, ...
   'Position',[804 43.5 114.75 27.75], ...
   'String','Compare', ...
   'Tag','Compare');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.831372549019608 0.815686274509804
0.784313725490196], ...
   'ListboxTop',0, ...
   'Position',[797.25 84 128.25 14.25], ...
   'String','Compare Measured to Theoretical', ...
   'Style','text', ...
   'Tag','Com');
if nargout > 0, fig = h0; end
```

3. Analyze_1.m

```
%
%  This m file is used as part of Kulite_Analysis
%
for i = 1:Rev;
    X(Loc(i)+1:Loc(i+1),(Num+3)) = X(Loc(i)+1:Loc(i+1),1) - time(i);
%translate time to distance
end
%
V = zeros(M,1);
for i = 1:Rev;
    V(i) = (pi*D)/(time(i+1) - time(i));     %velocity of each
revolution (in/s)
end
%
for i = 1:Rev;
    X(Loc(i)+1:Loc(i+1),(Num+4)) = V(i) * X(Loc(i)+1:Loc(i+1),(Num+3));
%distance around rotor for each rev (in)
end
%
%  Obtain Parameters from gui
%
div = findobj('Tag','divisions');
div = get(div,'String');
div = str2num(div);
%
num = findobj('Tag','Kulite');
num = get(num,'Value');
%
dev = findobj('Tag','STD');
dev = get(dev,'Value');
%
Ptot = findobj('Tag','Ptot_edit');
Ptot = get(Ptot,'String');
Ptot = str2num(Ptot);
%
Pamb = findobj('Tag','Pamb_edit');
Pamb = get(Pamb,'String');
Pamb = str2num(Pamb);
%
%  Obtain Slope and Intercept
%
load Slope.txt;
load Intercept.txt;
slope = Slope;
inter = Intercept;
```

```
%
%  Convert Voltage to Pressure
%
for i = 1:Rev;
   X(Loc(i)+1:Loc(i+1),(num+4+Num)) = (slope(num,1) *
X(Loc(i)+1:Loc(i+1),(num+1))) + inter(num,1);
end
%
%  Average Pressure For Each Division Around the Rotor
%
P = zeros(div+1,Num);
R = zeros(div+1,Num);
dis = [0:pi*D/div:pi*D]';
dis = dis*22/(pi*D);                    %Nondimensionalize the Rotor
%
X(:,Num+4) = (div/(pi*D)).*X(:,Num+4);  %Assign Distances to a Division
X(:,Num+4) = round(X(:,Num+4));         %Round to Nearest Division
%
d = X(Loc(1)+1:Loc(Rev+1),Num+4);
cla
%
for i = 0:div;
   k = find(d==i);
   k = k + Loc(1);
   A = size(k,1);
   if A==0;
      plot(0,0)
      text(-.15,0,'Number of divisions is too large')
      error('Number of divisions is too large')
   else
   end
   p = zeros(A,1);
   for j = 1:A;
      p(j,1) = X(k(j),num+Num+4);
      p(j,1) = p(j,1) + Pamb;          %Convert to absolute pressure
      p(j,1) = p(j,1)/(Ptot*1.4);      %Nondimentionalize the pressure
   end
   R(i+1,1) = std(p);            %Standard Deviation
   P(i+1,1) = mean(p);           %Average Pressures of Each Division
end
%
RUp = P + R;
RDwn = P - R;
%
%  Plot
%
if dev == 1;
   plot(dis,P,'.');                        %plot pressure vs. distance
   hold on
   plot(dis,RUp,'xr');
   plot(dis,RDwn,'xk');
   line([dis';dis'],[RDwn';RUp']);
else
   plot(dis,P,'.');                        %plot pressure vs. distance
end
temp = axis;
temp(1,2) = 22;
```

```
axis(temp);
xlabel('Non-dimensionalized Distance Over The Rotor');
ylabel('Pressure Ratio (Pabs/(Pt*gamma))');
title('Unsteady Pressure Measurements Over A Transonic Compressor
Rotor');
```

4. Analyze_2.m

```
%
%  This m file is used as part of Kulite_Analysis
%
cla
points = div/22;                   %Number of pressure points per blade
pitch = [1/points:1/points:1]';    %Pitch along averaged blade
averbl = zeros(points,1);
r = zeros(points,1);
%
for i = 1:points;
   pr = zeros(22,1);
   for j = 0:21;
      pr(j+1,1) = P(i+(j*points),1);
   end
   r(i,1) = std(pr);
   averbl(i,1) = mean(pr);
end
%
rup = averbl + r;
rdwn = averbl - r;
%
Dev = findobj('Tag','std');
Dev = get(Dev,'Value');
%
%  Plot
%
if Dev == 1;
   plot(pitch,averbl,'.');                 %plot pressure vs. distance
   hold on
   plot(pitch,rup,'xr');
   plot(pitch,rdwn,'xk');
   line([pitch';pitch'],[rdwn';rup']);
else
   plot(pitch,averbl,'.');                 %plot pressure vs. distance
end
temp = axis;
temp(1,2) = 1;
axis(temp);
xlabel('Pitch');
ylabel('Pressure Ratio (Pabs/(Pt*gamma))');
title('Unsteady Pressure Measurements Over A Transonic Compressor
Rotor');
save Average.txt averbl -ascii;
save Pitch.txt pitch -ascii;
```

5. Analyze_3.m

```
%
%  This m file is used as part of Kulite_Analysis
%
cla
[newfile,newpath] = uigetfile('*.txt','Select File');
TXT.filename = [newpath newfile];
theo = load(TXT.filename);
plot(theo(:,1),theo(:,2),'b');
hold on
%
points = div/22;                   %Number of pressure points per blade
pitch = [1/points:1/points:1]';    %Pitch along averaged blade
averbl = zeros(points,1);
r = zeros(points,1);
%
for i = 1:points;
   pr = zeros(22,1);
   for j = 0:21;
      pr(j+1,1) = P(i+(j*points),1);
   end
   averbl(i,1) = mean(pr);
end
%
[Q,I] = max(theo);
theohigh = I(2);
[Q,I] = min(theo);
theolow = I(2);
theomean = .5*(theohigh + theolow);
theomean = floor(theomean);
[e,f] = size(theo);
bigpitch_theo = theo(:,1)*100;
bigpitch_theo = round(bigpitch_theo);
theopitch = bigpitch_theo(theomean,1);
%
[Q,I] = max(averbl);
meashigh = I;
[Q,I] = min(averbl);
measlow = I;
if measlow < meashigh;
   measlow = measlow + points;
end
measmean = .5*(meashigh + measlow);
measmean = floor(measmean);
if measmean > points;
   measmean = measmean - points;
end
[g,h] = size(averbl);
bigpitch_meas = pitch*100;
bigpitch_meas = round(bigpitch_meas);
measpitch = bigpitch_meas(measmean);
middle = 100*theopitch/points;
middle = round(middle);
%
```

```
pitch = [1/points:1/points:1]';
if measmean <= middle;
   plot(pitch(middle:g),averbl(measmean:(measmean+g-middle)),'b.')
   plot(pitch(1:(middle-measmean)),averbl((measmean+g-
middle+1):g),'b.')
   plot(pitch((middle-measmean+1):(middle-1)),averbl(1:(measmean-
1)),'b.')
else
   plot(pitch(middle:(g-measmean+middle)),averbl(measmean:g),'b.')
   plot(pitch((g-measmean+middle+1):g),averbl(1:(measmean-
middle)),'b.')
   plot(pitch(1:(middle-1)),averbl((measmean-middle+1):(measmean-
1)),'b.')
end
temp = axis;
temp(1,2) = 1;
axis(temp);
xlabel('Pitch');
ylabel('Pressure Ratio (Pabs/(Pt*gamma))');
title('Comparison of Average Measured Blade to Theoretical Unsteady
Pressure');
legend('34% Axial Chord','Measured Pressure');
save Average.txt averbl -ascii;
save Pitch.txt pitch -ascii;
```

APPENDIX F: STEEL CASE WALL: KULITE INSTALLATION DESIGN

A steel case wall was designed and built for the transonic compressor. The case wall was to ensure better tip clearance accuracy than could be obtained with a Plexiglas case wall. Instrumentation to obtain the pressure map over the blade tip, for comparison with prediction, was designed into the new steel case wall. Six Kulite pressure transducers were thought to be needed to adequately resolve the pressure map, one upstream, one downstream, and four across the blade. The calibration and data analysis MATLAB scripts were written for six Kulites. The installation holes for the Kulites, and corresponding static pressure taps were designed and drilled, and the aluminum slugs to hold the Kulites, were manufactured. The steel case wall, along with the Kulite pressure transducers, will be installed after the transonic compressor reaches 100% design speed with the Plexiglas case wall.

A. INSTALLATION OF PROBES

1. Method of Installation

Figure 79 shows the slug designed to hold the Kulite pressure transducer in the steel case wall. The slug was based on the Vavra design however it was made longer to account for the epoxy trench built into the steel case wall. The epoxy trench provided an abradable liner to protect the rotor in case of rubs. The slug is also longer for ease of retracting the pressure transducer from the case wall and to help protect the Kulite and it’s fragile wires. Solid slugs were also produced to install in the case wall, when the Kulite is not required.

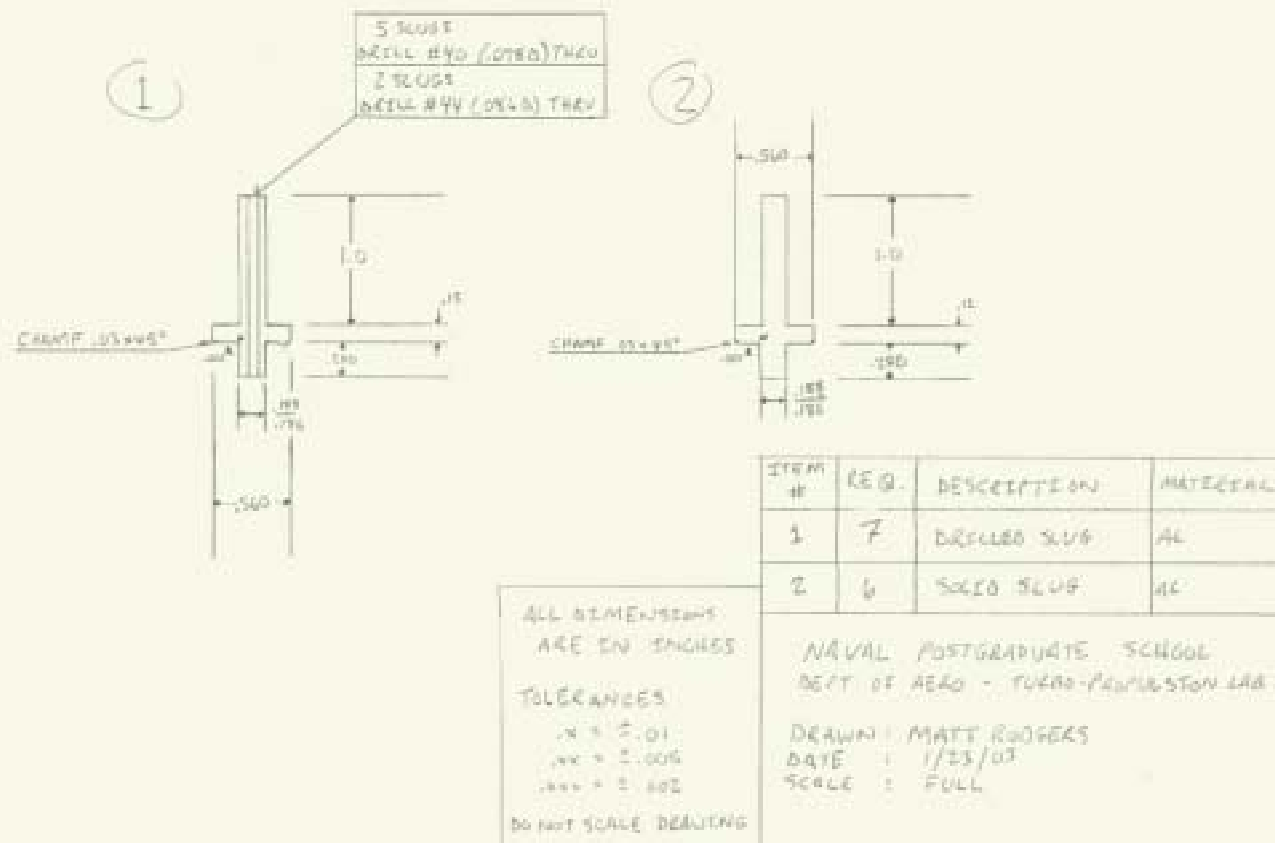


Figure 79. Steel Case Wall Kulite Mounting Design

2. Location of Probes

Figure 80 shows the placement of the Kulite pressure transducers and corresponding static pressure taps in the steel case wall. The axial locations correspond, and are given in the figure. The Kulite and corresponding static pressure tap locations were spaced one blade spacing apart. The Kulite and static pressure tap locations, across the blade, were placed at 10.5%, 37%, 63%, and 89.5% axial chord.

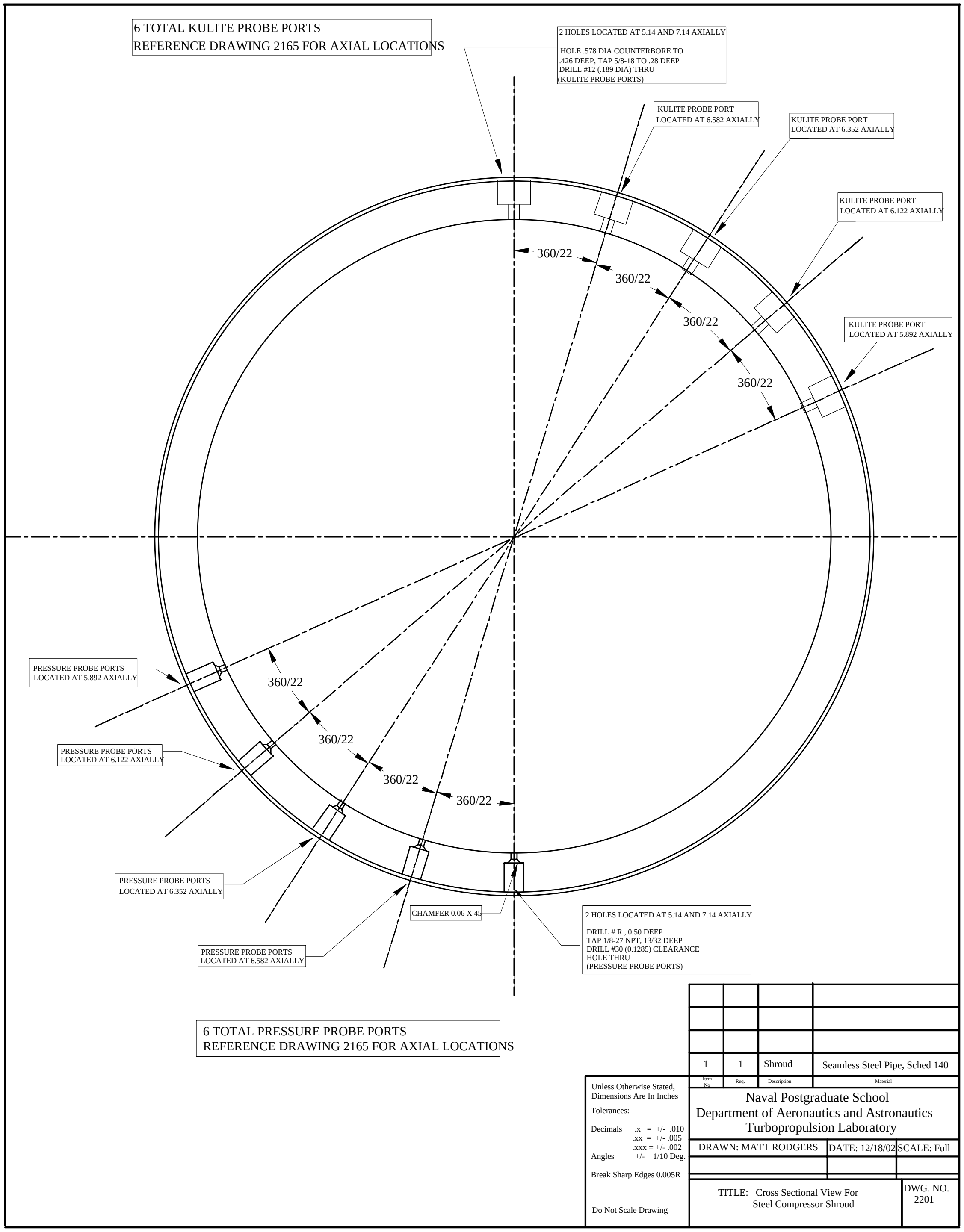


Figure 80. Kulite Installation in Steel Case Wall

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Paige, G.C, Measurement of Case Wall Pressure Signatures in a Transonic Compressor Using Real-Time Digital Instrumentation, Master's Thesis, Naval Postgraduate School, Monterey, California, June 1976.

2. O'Brien, J.M., Transonic Compressor Test Rig Rebuild and Initial Results with the Sanger Stage, Master's Thesis, Naval Postgraduate School, Monterey, California, June 2000.

3. Sanger, N.L., Design of a Low Aspect Ratio Transonic Compressor Stage Using CFD Techniques, ASME Journal of Turbomachinery, Vol. 118, pp. 479-491, July 1996.

4. Hobson, G.V., (Unpublished) Department of Aeronautics and Astronautics, Naval Postgraduate School, Monterey, Calfornia, May 2003.

5. Erwin, J. R., A Review of the Design of the NPS/TPL Transonic Compressor, Contractor Report No. NPS67-83-004CR, Naval Postgraduate School, Monterey, California, 1983.

6. Kulite Semiconductor Products Incorporated, Kulite Miniature IS Silicon Pressure Transducers, One Willow Tree Road, Leonia, NJ 07605, April 1995.

7. Hewlett-Packard Operating Manual, HP E1422A Remote Channel Multi-function DAC Module, Edition 1, E1422-90000, May 1999.

8. HP VEE Pro (version 6.01), August 2000

9. Hewlett-Packard Operating Manual, HP E8402A, E8404A C-Size Mainframes, Edition 1, E8402-90000, May 1998.

10. Agilent Technologies Operating Manual, N2216A VXI/SCSI Interface Module, N2216-90001, July 2000.

11. HP DAC Express (version 2.01), September 2000

12. MATLAB (version 5.3.0 R11), January 1999

13. Chima, R.V., Calculation of Tip Clearance Effects in a Transonic Compressor Rotor, ASME Journal of Turbomachinery, Vol. 120, pp. 131-140, January 1998.

14. Chima, R. V., Calculation of Multistage Turbomachinery Using Steady Characteristic Boundary Conditions, AIAA-98-0968 or NASA/TM-1998-206613.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, VA

2. Dudley Knox Library
Naval Postgraduate School
Monterey, CA

3. Chairman and Distinguished Professor Maximilian F. Platzer, Code AA/PL
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, CA

4. Professor Raymond P. Shreeve, Code AA/SF
Department of Aeronautics & Astronautics
Naval Postgraduate School
Monterey, CA

5. Professor Garth Hobson, Code AA/HG
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, CA

6. Naval Air Warfare Center – Aircraft Division
Propulsion and Power Engineering
ATTN: R. A. Ravindranath, Code 4.4.7, BLDG 106
Patuxent River, MD

7. Naval Air Warfare Center – Aircraft Division
Propulsion and Power Engineering
ATTN: M. Klein, Code 4.4.7.1, BLDG 106
Patuxent River, MD

8. Naval Air Warfare Center – Aircraft Division
Propulsion and Power Engineering
ATTN: C. Gorton, Code 4.4.7.1, BLDG 106
Patuxent River, MD

9. Mr. Nelson Sanger
Rocky River, OH

10. Matt Rodgers
Monterey, CA